UNIVERSIDAD DE LA RIOJA

TRABAJO FIN DE ESTUDIOS

Título

Estudio y desarrollo de sistemas de identificación basados en NFC

Autor/es

DAVID GONZÁLEZ FABIÁN

Director/es

JOSÉ JAVIER MARTÍNEZ SANTOLAYA

Facultad

Escuela Técnica Superior de Ingeniería Industrial

Titulación

Grado en Ingeniería Electrónica Industrial y Automática

Departamento

INGENIERÍA ELÉCTRICA

Curso académico

2019-20

publicaciones.unirioja.es
E-mail: publicaciones@unirioja.es

UNIVERSIDAD DE LA RIOJA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL

TRABAJO DE FIN DE GRADO

TITULACIÓN: Grado en Ingeniería Electrónica Industrial y Automática

CURSO: 2019/2020 **CONVOCATORIA: SEPTIEMBRE**

TÍTULO:

Estudio y desarrollo de sistemas de identificación basados en NFC

ESTUDIANTE: David Gónzalez Fabián

TUTORES/AS: JOSÉ JAVIER MARTÍNEZ SANTOLAYA

DEPARTAMENTO: Ingeniería Eléctrica

Resumen

En este proyecto se realiza un estudio sobre los sistemas de identificación que utilizan tecnologías de radio frecuencias, y centrándose en concreto en las tecnologías relacionadas con el protocolo de radiofrecuencias conocido como NFC, o comunicación de campo cercano. El principal objetivo de este estudio es entender cómo funciona la tecnología, para tener la certeza de la creación de un sistema que es seguro desde sus cimientos, y así poder prever ataques y filtrados de información sensible, como serian claves del sistema o datos personales.

El estudio se comienza con un análisis de la tecnología de identificación de RFID, cómo funciona, origines de esta tecnología, y las distintas técnicas y protocolos que se utilizan para la seguridad, además de los distintos ataques y vulnerabilidades, así como las aplicaciones comunes de esta tecnología.

Posteriormente se estudia el NFC, tecnología basada en RFID, que incorpora protocolos propios y que ha sido adoptada por grandes empresas como estándar de tarjetas de autentificación y sistemas de pago.

Por último, se expone el desarrollo de un sistema de identificación NFC, usando tecnologías y estándares diseñadas en el último año que cuentan con los últimos avances en seguridad.

Abstract

In this Project an study about the RFID, Identification with Radio Frequency is made, and its focused mainly in the technologies related the NFC Protocol, also known as Near Field Comunication. The main objective of this study is learning and understanding how this technology works, so you can have the assurance of the creating of a secure system were the security comes from a root-of-trust. This way attacks and sensible information leaks can be prevented, like system keys or personal data.

This study starts with an analysis of the RFID technology, how its works, its roots, the different techniques and protocols used to obtain this security, as well the different attacks and vulnerabilities this technology can face, as well as it different applications.

After that the NFC technology is studied, which is a RFID-based technology, that has it own protocols and that big companies have adopted, as a standard in authentication and payment cards.

At last, a development of an identification NFC system is exhibit, using state-of-the-art technology and protocols, designed in the last year.

1 Índice general

Índice de Figuras

2 Memoria

Presentación

Este trabajo de Fin de Grado se presenta como objeto de la obtención del Título de Graduado en Ingeniería Electrónica Industrial y Automática por la Escuela Técnica Superior de Ingenierías de la Universidad de La Rioja.

Realizado por David González Fabian, en el año 2020.

Objeto

El objetivo de este trabajo de fin de grado es el estudio de la tecnología NFC, y su implementación en un sistema de autentificación que permita, a partir de una o varias tarjetas nuevas, incorporarlas al sistema y permitirles acceder, ya sea a una sala o a una base de datos.

Este sistema de acceso ha de ser completamente seguro desde la creación de las tarjetas de acceso hasta la comunicación de estas, siguiendo protocolos y tecnología modernas.

Estimamos que un sistema es seguro, si toda la comunicación y transacciones de información se realizan de manera completamente cifrada, y si las claves que cifran esa información se inyectan y almacenan de forma segura.

Implementar en un prototipo un modelo de autentificación comercial, usando librerías y entendiendo que se hace en cada momento, para tener la certeza de esa seguridad.

Objetivos parciales y tareas

Para poder darle sentido y valor a la implementación de un sistema de autentificación mediante NFC, se han seguido estas tareas u objetivos, reflejadas en la memoria, con el fin de entender lo que ocurre en una autentificación segura.

Análisis de la Tecnología RFID y NFC

Estudio del Funcionamiento de la Comunicación NFC

Estudio y comprensión de la criptografía que existe detrás de los protocolos NFC, y los protocolos en sí.

Estudio de las tecnologías modernas, microcontroladores y placas de desarrollo para implementar una aplicación NFC de autentificación.

Estudio, compresión y síntesis de las librerías usadas por esas tecnologías.

Selección de librerías y tecnologías.

Implementación de las librerías en un entorno de desarrollo, con llamadas a funciones y modificaciones para lograr el objetivo de autentificación.

Análisis de accesorios o mejoras al sistema de autentificación planteado.

Justificación

La tecnología NFC e se utiliza en el día a día, y cada día más utensilios y elementos tienen conectividad NFC incorporada ya que en muchas ocasiones genera un valor añadido a este objeto sin necesidad de apenas alterar su coste.

Esta tecnología autentifica, comunica y sirve como dispositivo de pago, lo que la hace susceptible de muchos ataques. La implementación de criptografía que se utiliza alrededor de las comunicaciones entre dispositivos avanza, se modifica y se reinventa. Y los dispositivos NFC varían con estos avances.

En todo el mundo se utilizan sistemas de identificación o autentificación por NFC, y si el sistema está mal implementado, o utiliza tecnología obsoleta, el recinto o objeto cuya seguridad dependa del NFC, es vulnerable.

El concepto de este TFG desde esta premisa, es la creación e implementación de un caso de estudio de autentificación, donde la tecnología NFC sirva como método de autentificación y que además, el diseño e implementación de esta tecnología sea la opción más segura posible dentro del ámbito del NFC comercial.

Esquema de los apartados del Documento.

La memoria se encuentra dividida de esta forma:

Se comienza con una explicación de la tecnología RFID, su funcionamiento de forma física y aplicaciones.

Continuando el apartado anterior, explicación de la tecnología NFC, sus protocolos, su sistema de comunicarse, amenazas, y aplicaciones.

Dentro del NFC, el TFG se centra en el estudio de la seguridad de esta tecnología, explicando la criptografía que se utiliza en los mensajes y los protocolos de comunicación segura.

Una explicación del caso práctico, donde se usa la tecnología de NXP para, junto con un elemento que le da valor añadido de seguridad a la comunicación NFC. El caso práctico está dividido en una explicación sencilla, nada segura y el sistema completo con seguridad.

Por último, el planteamiento de un caso extra como mejora donde una persona que tenga un certificado tuyo de AWS, pueda acceder a la casa en un horario determinado.

Cronograma

Cronograma de Actividades

#	Activity			Comienzo	Fin	SEMANAS 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
20	**Total del Proyecto**	01/02/20	03/09/20	0	31	
40	**Diseño de Prototipo (125h)**	01/04/20	20/08/20	9	29	
41	Revision de conceptos (30h)	01/04/20	15/04/20	9	11	
42	selección de tecnologias (15h)	15/04/20	15/05/20	11	15	
43	Selección Final + librerias (80h)	20/06/20	20/08/20	20	29	
60	**Realizacion de la documentacion(160h)**	01/02/20	22/08/20	0	29	
61	Lectura de Libros (80h)	01/02/20	15/04/20	0	11	
62	Resumenes de puntos importantes (10h)	01/03/20	20/06/20	4	20	
63	Escritura de memoria(70h)	22/06/20	22/08/20	20	29	
80	**Arreglos y optimizaciones (20h)**	15/07/20	01/09/20	24	31	
100	**Modificaciones al Codigo (50h)**	23/06/20	14/08/20	21	28	
101	Codigo General (20h)	23/06/20	01/08/20	21	26	
102	Modificacion de Librerias(30h)	10/07/20	14/08/20	23	28	

Normativas.

Estas son las normativas por las que se rige el caso práctico estudiado en el proyecto. Existen muchas más normas relacionadas con las tecnologías RFID y NFC, el resto de normas están mencionadas en anexos.

Estas normas son.

ISO:

ISO 15692 – Explica el protocolo de intercambio de datos en los sistemas RFID.

ISO/IEC 15693, Otra normativa, las tags que utilizan esta especificación suelen usarse para aplicaciones destinadas a la industria, como el etiquetado inteligente. Un ejemplo de chips de esta tecnología son los ICODE de NXP. Se considera tecnología contactless porque usan 13.56MHz.

ISO/IEC 14443, dividida en cuatro partes, definida para NFC

La primera define las propiedades físicas y las medidas de las tarjetas contactless.

La segunda define la frecuencia y los dos tipos de modulación de señal.

La tercera describe el proceso de comunicación, centrándose sobre todo en los comandos y cómo lidiar con las colisiones (varias tarjetas en el mismo campo)

La cuarta especifica protocolo de transmisión y de autentificación.

ISO 10536,11784,11785 – Definen el estándar de RFID, estructura y manejo de datos.

ISO 18000 – Define las distintas frecuencias y las interfaces y parámetros genéricos de 7 frecuencias globalmente aceptadas. Se complementa con ISO 24710 y 24729 para usarse en industria o tecnologías en concreto.

ISO 18092 – Describe la interfaz y protocolo del NFCIP-1

NFC Forum.

El foro de nfc especifican los tipos de tag (tarjeta) y la normativa de intercambio de datos NDEF. Existen 5 tipos de tarjeta, en función de que normativa detallada dentro de la ISO 14443 sigan.

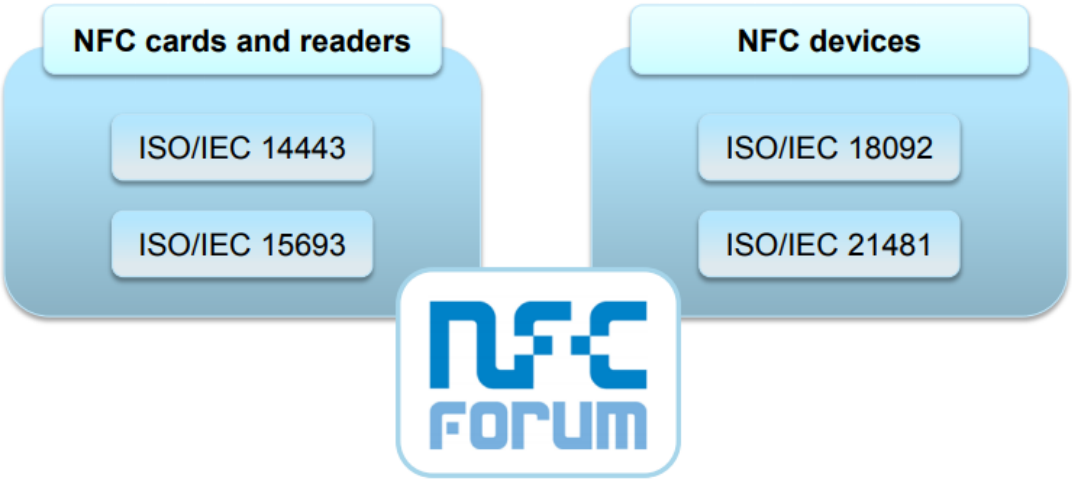


Figura 1 Estándares ISO por los que se rige el NFC forum

Conocimientos Previos

En los últimos años los procedimientos relacionados con la identificación automática (auto ID) se han popularizado en muchos sectores, como la logística de venta y distribución, la industria, las compañías de manufacturas y más. Los procedimientos de identificación se usan también para dar información sobre personas, usuarios, animales y productos en tránsito o fabricación.

Los códigos de barras son un ejemplo de etiquetas que significaron una revolución en cuanto a sistemas de identificación en productos, pero que en la actualidad no son eficientes para algunos casos. Los códigos de barras son muy baratos, pero apenas pueden contener datos y no pueden ser reprogramados.

La solución óptima para almacenar datos es dentro de un microchip. Un ejemplo son las tarjetas Smart, que funcionan con contacto mecánico. Este chip se encuentra en tarjetas de crédito, tarjetas de identificación como el DNI y en tarjetas SIM del teléfono móvil. En muchas ocasiones, este tipo de contacto mecánico es muy impráctico y muestra muchas desventajas debido a fallos de contacto (sabotaje, tierra o polvo, desgaste de la cobertura). Una transferencia de datos entre el dispositivo que almacena los datos y el lector es mucho más flexible y útil. En un caso ideal, la energía necesaria para que el dispositivo que almacena los datos funcione sea suministrada también desde el lector usando tecnología sin contacto. El nombre que reciben estas tecnologías es el de RFID.

Los británicos emplearon sistemas que serían la base del RFID durante la Segunda Guerra Mundial para identificar sus propias aeronaves (IFF, Friendo or Foe Identificación). Tras la guerra, varios laboratorios trabajaron con esta tecnología en lo que serían los primeros sistemas de identificación de personas, con el fin de limitar el acceso a diversas áreas y crear IDs más difíciles de falsificar. Otros sectores comenzaron a usarlas, como la identificación de equipaje en aeropuerto y los juguetes interactivos DGT tableros de ajedrez). (Want, 2006)

Su uso no se estandarizo hasta que en los años noventa tres grandes organizaciones decidieron usar esta tecnología en gran escala. Estas fueron Wal-Mart, Tesco y el departamento de defensa de Los Estados Unidos. En esta época el chip RFID era de un tamaño similar a una tarjeta de crédito, mientras que en la actualidad existen chips más pequeños que el mm^2

Figura 2 Logo del RFID

La participación del gobierno de estados unidos también supuso algunos avances en esta tecnología. El departamento de Energía (DOE) desarrollo un sistema para el seguimiento de desechos nucleares. Instalaron etiquetas de RFID en los camiones y lectores en las salidas de los garajes. Este mismo sistema se comercializo en las décadas posteriores y se sigue utilizando en túneles y peajes por todo el mundo. El departamento de Agricultura de los estados unidos (USDA) desarrolló RFID pasivos para el seguimiento del ganado vacuno, para distinguir que vacas habían sido medicadas y cuales necesitaban una dosis de medicación aún.

Junto con el uso de bases de datos online y una red de comunicaciones (llamados sistemas cyber-físicos o CPS), estos podían identificar el estado y posición de cualquier producto mientras se movían entre fábricas, vehículos hasta el almacén de cada tienda. Esto se llevó a cabo y dio lugar en 1999 a la formación del Auto ID center, un consorcio de empresas y equipos científicos destinados a investigación y estandarización de sistemas modernos y asequibles de autoidentificación. En 2004 el Auto-ID center paso a llamarse EPCglobal, que creo varios estándares de identificación automática.

Phillips y Sony trabajaron en protocolos propios de tecnología contact-less, dando lugar a Mifare por parte de Phillips y FeliCa de Sony. Esto dio lugar a que se estableciera el protocolo NFC, aprobado como estándar ISO 18092

En el 2004, Phillips, Sony y Nokia crearon el NFC FORUM, que se encargó de del desarrollo y especificaciones de esta tecnología, a la que apoyaron empresas como Google, Visa y PayPal, entre otras.

En el año 2006 se comenzaron a realizar pagos mediante NFC y aparecieron los primeros móviles con antenas NFC incorporadas. En 2011 apareció Google

Wallet, que permitía la digitalización de tarjetas de crédito y realizar pagos con ella a través de teléfono.

Tras acabar la década, el valor de mercado de RFID era de $900 Millones de USD.(Krebs, n.d), convirtiéndolo en el sector de la industria de radio tecnología que más rápido crecía, por encima de los teléfonos móviles.

En la actualidad, el mercado de RFID vale $11.0 miles de millones de USD, y se estima que esa cifra alcanzara $13.4 en 2022. ***(IDTechEx)***

RFID (Radio Frequency Identification)

La identificación por radiofrecuencia se refiere a la tecnología que utiliza ondas de radio para identificar *Tags o transpondedores (Etiquetas RFID)* a una distancia determinada. Estas Tags son transmisores que envían ráfagas de información a través de una antena. Existen principalmente 3 tipos distintos de tags. Activos, Pasivos y Semi-activos.

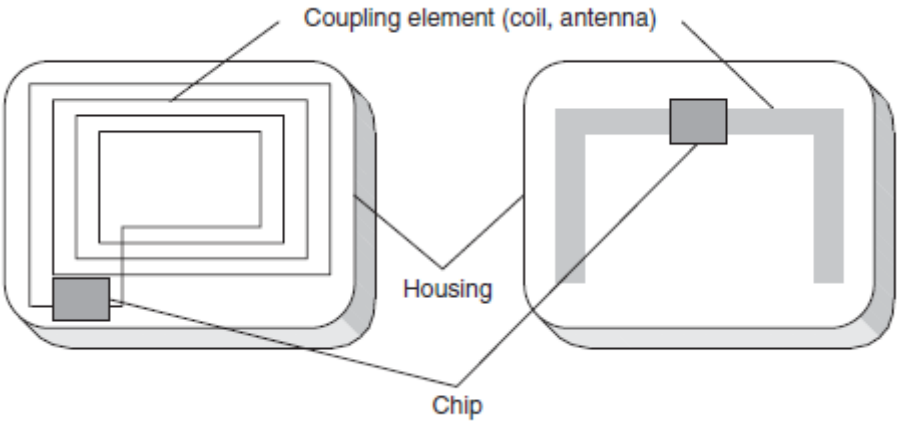


Figura 3 Chip con antena pasivo RFID

Los tags activos usan una batería lo que les permite tener la mayor distancia de funcionamiento, pero implica un coste mayor. Esta batería no da energía a la transmisión de datos, ya que sirve de manera exclusiva para darle energía al microchip y para la retención de datos. La energía del campo electromagnético es la única fuente de energía usada para la transmisión de datos entre tag y lector. Al llevar una batería pueden estar unidos a sensores adicionales, por ejemplo, para identificar el estado del contenedor al que están unidos.

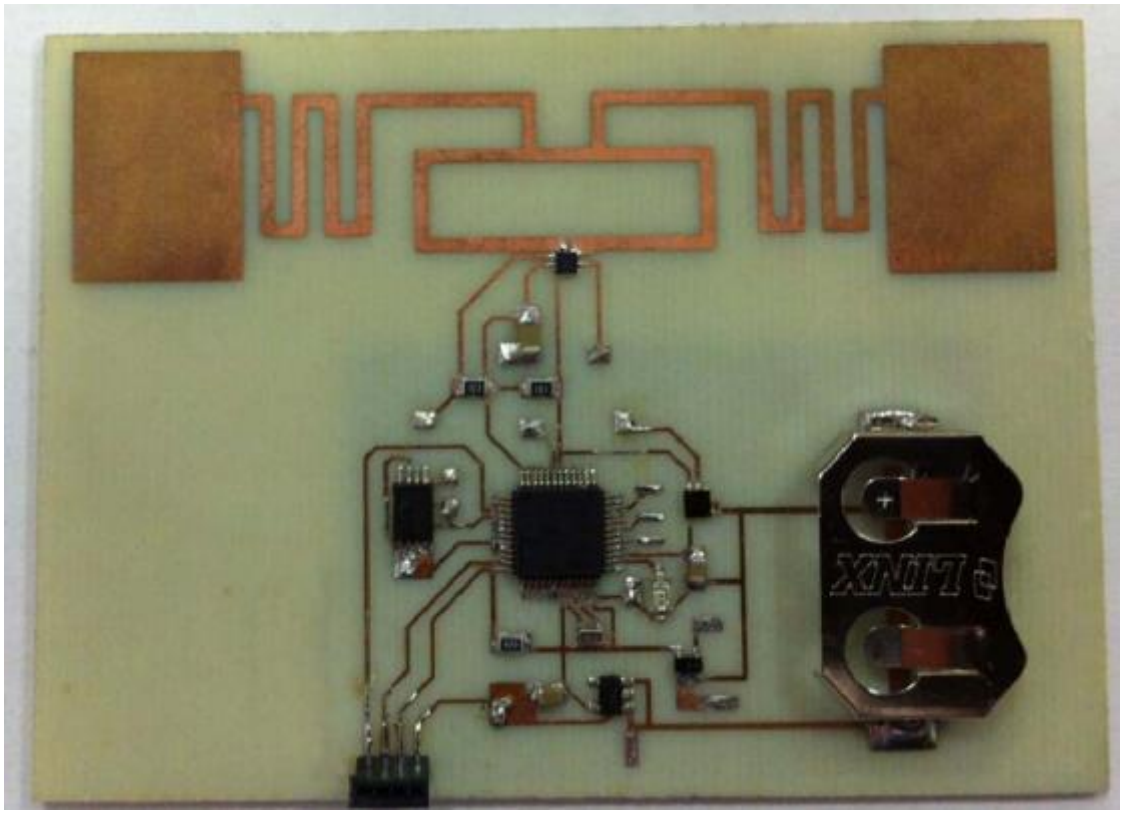


Figura 4 PCB de Tag Activa de RFID

Los pasivos utilizan el campo magnético que genera el lector como fuente de energía, usando la inducción magnética, para transmitir su información y para dar energía a el microchip, debido a la ausencia de batería, tienen un coste más reducido y un tiempo de uso mayor, pero un límite de distancia mayor.

Los Semi-activos usan una batería para extender la distancia de lectura y para reducir las anomalías de lectura de datos.

Dentro de los tags pasivos podemos distinguir entre tags sencillos (low-end) y más complejos (high-end).

Los sencillos son del tipo Read-only, en ocasiones conocidos también como etiquetas inteligentes (Smart labels).

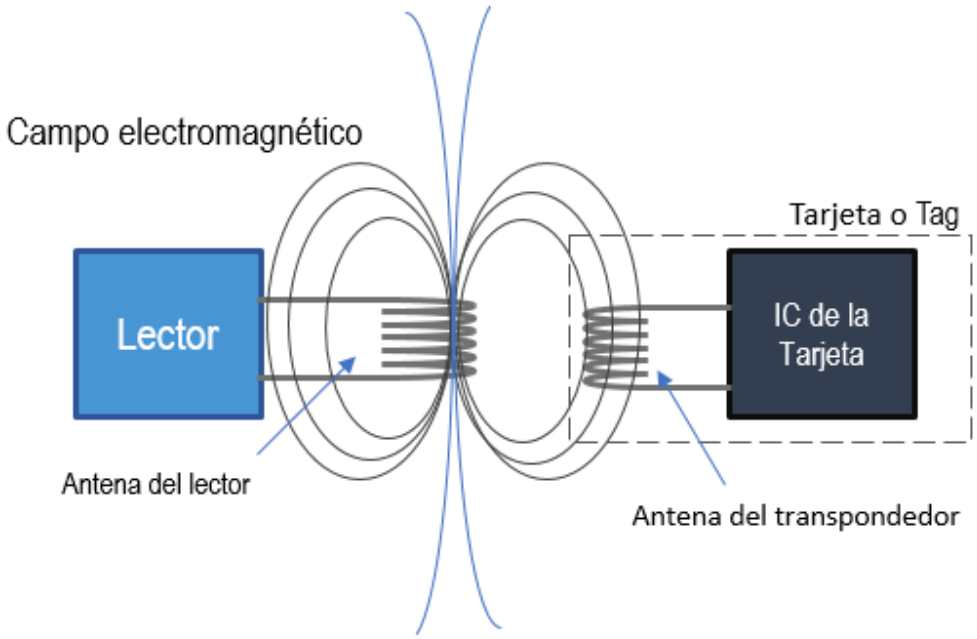


Figura 5 : Ejemplo de unión de Lector y tag pasiva de RFID

Estos tienen codificados un número de serie único de varios bytes, establecido por el EPC como 12 bytes, pero pueden tener una mayor capacidad, son en cierta forma un sustituto de los códigos de barras. Si se coloca cerca de un lector de Radio Frecuencias, este emitirá de manera continua su este número. El lector no puede comunicarse con el tag, y tan solo puede haber un único tag en la zona del lector, para evitar colisiones de datos. En la actualidad este tipo de tags se utilizan debido a que al ser más simple el área del chip es muy reducida, lo que implica además un coste reducido de manufacturación. Un ejemplo de esto es la identificación de animales [ISO 11785]

Estos tags de identificación para animales se comercializaron en la década de los 90 con una frecuencia de 125 kHz. En la actualidad se utilizan tags de una frecuencia superior (13.56 MHz), que tienen un mayor rango de alcance y de velocidad de transferencia.

Los más complejos o high-end tienen una memoria en la cual, si que se puede escribir mediante un lector, estos se conocen en el mercado como tags reescribidles. Son capaces de procesar comandos sencillos para leer datos en concreto, reemplazar información, procedimientos anticolisión para evitar errores en la lectura si hay varias etiquetas en la zona. Suelen almacenar certificados de servidor, que se modifican cada vez que se conectan con un lector. Son algo más seguras, pero siguen siendo vulnerables a varios ataques que puedan clonarla o seguirla (tracking) (Sanjay Ahuja, 2010)

Algunos de los más complejos combinan microprocesadores y un sistema con una Smart card. Con el uso de microprocesadores se realizan los cálculos de

Estos tags de identificación para animales se comercializaron en la década de los 90 con una frecuencia de 125 kHz. En la actualidad se utilizan tags de una frecuencia superior (13.56 MHz), que tienen un mayor rango de alcance y de velocidad de transferencia.

Además del circuito integrado o chip, el tag RFID está formado por una antena. Aunque la más común es rectangular, existen varias formas de colocar la antena. Las funciones de la antena son: absorber las ondas de radio frecuencia para usarlas como alimentación y difundir los datos contenidos en el chip en esa misma frecuencia. El tamaño de la antena delimita tanto la distancia de transmisión como la complejidad del chip. A mayor sea el tamaño de la antena, mayor es la energía que puede adquirir del campo y mayor es la potencia a la que puede transmitir y que puede usar para alimentar el circuito integrado.

El rango de frecuencia a la cual puede operar los sistemas de RFID abarca desde 125kHz (onda larga) hasta 5.8GHz(microondas). La distancia de lectura puede estar por debajo de los milímetros hasta más de 20m. (Peter Darcy)

Los campos generados pueden ser magnéticos (LF y HF, frecuencia baja y alta), o electromagnéticos (UHF, Ultra High Frequency). A los magnéticos, que formas más del 90% del mercado de RFID se les conoce como sistemas inductivos. Dentro de estos sistemas hay varios estándares, que definen los parámetros técnicos de tanto el lector como el tag para varias aplicaciones, como las Smart cards, identificación de animales o automatización industrial. [ISO 15693]

Los sistemas que superan la distancia de 1 metros se conocen como sistemas de rango largo y funcionan utilizando ondas electromagnéticas en el rango de frecuencia “ultra alta” UHF (868MHz) y microondas (2.5GHz). Hasta 3 metros se pueden alcanzar con tags pasivos mientras que con activos se pueden alcanzar la distancia de 3 kilómetros.

Hoy en día es esta frecuencia, 13.56 MHz la que se ha estandarizado y se utiliza en sistemas de control de acceso y sistemas de pago.

El Lector de RFID descubre los tags que se encuentren dentro del rango de proximidad del lector. Si se descubre, la ID y la información contenida en el tag se transmitirá a través de la antena y será leída. Tras ser leída, el microcontrolador del lector se encargará de la seguridad y comprobará la fiabilidad de esos datos. Dependiendo del uso de las Tags, la información puede ser procesada directamente por el lector o enviada a una base de datos donde se almacene.

El campo de radio frecuencia que genera el lector tiene tres propósitos, si se comunica con tags pasivas.

Inducir la suficiente energía en la antena del tag, proporcionarle una fuente de clock al tag, y servir como portador para devolver la información que emite la tag.

Esto se conoce como modulación por retrodispersión. El funcionamiento es el siguiente. El lector genera una onda senoidal, y observa si se produce algún tipo de modulación en esta onda. Si se produce esta modulación es indicativo de la presencia de un tag en el sistema.

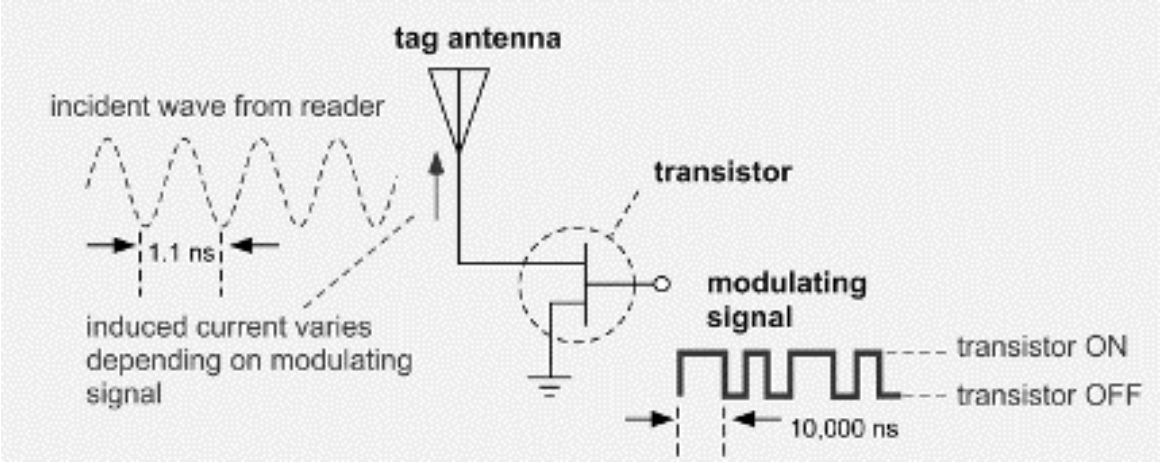


Figura 6 Modulación de señal

El tag entonces absorbe energía de este campo generado por el lector. Envía sus datos a ritmo de reloj, usando la onda senoidal, a un transistor que se encuentra conectado a la antena.

Este transistor deriva la antena, siguiendo la secuencia que le ha enviado el chip. Esta derivación genera una fluctuación de la onda portadora, lo que genera un ligero cambio en la amplitud de la onda.

Finalmente, el lector detecta los cambios en la amplitud de la onda, y genera un flujo de bits acorde, habiendo leído la información del tag.

El funcionamiento electro es muy similar al de un transformador. Hay dos “embobinados” el del lector y el del tag. Cuando el embobinado secundario, el del tag se ve fluctuado por el transistor, el primer embobinado detecta una caída o subida de voltaje.

Existen muchas formas de modular la señal y los datos, en función de los ciclos, los distintos protocolos de comunicación y de codificación de datos y se han ido variando con los años. Más adelante se explicará el pertinente a el apartado de la tecnología NFC, pero este estudio habla de los distintos algoritmos para codificación de datos y algunos estándares de modulación.

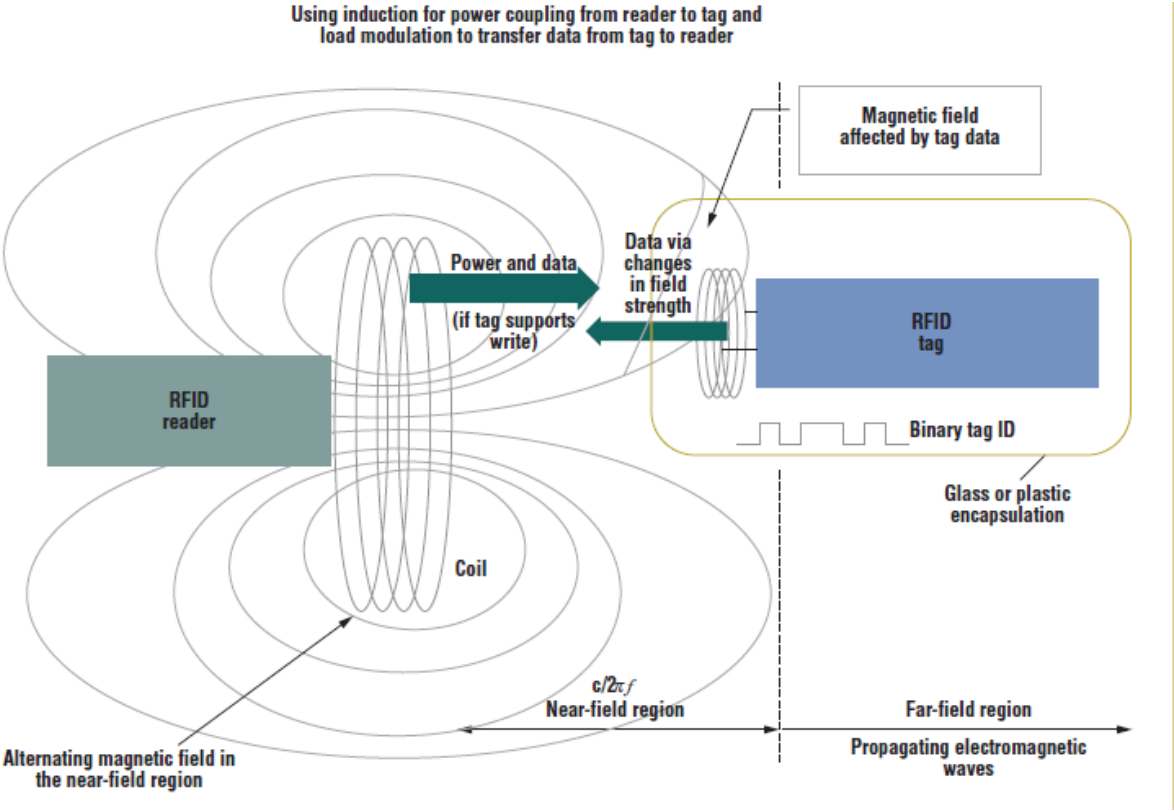


Figura 7 Induccion de los embobinados de lector y tag

Existen dos tipos distintos de lectores, en función de la cantidad de bobinas que tengan. Los lectores simples o de bobina única, en la cual es la misma bobina la que transmite energía y datos, y los lectores de dos bobinas, en la cual una transmite energía y la otra transmite datos. Los lectores han de ser capaces de acondicionar la señal que reciben del tag, y corregir los errores y evitar la colisión de información.

Normalmente lector y etiqueta no forman parte de un sistema aislado, sino que están unidos a una base de datos o a una red. El Middleware se encuentra situado entre todo el software de la base de datos del cliente y el hardware del lector de RFID. Sus funciones principales son la de gestionar la información que se envía y se recibe de la etiqueta, y transmitir solo la información útil al software. En muchas ocasiones, se encarga también de la encriptación y lectura de datos.

Debido a los beneficios de esta tecnología frente a otras tecnologías de autoidentificación el RFID se utiliza en una gran variedad de sectores y de maneras muy diversas. Ejemplos de esto son hospitales en los cuales los pacientes que necesitan cuidado máximo llevan pulseras con tags incorporadas. En algunos aeropuertos, se usan tags con el equipaje para asegurarse que se sabe de su posición en todo momento. A las mascotas se les implantan chips

RFID para asegurarse que, en caso de perderse, las autoridades puedan encontrar la información de sus dueños simplemente escaneando el tag. **(Finkenzeller, 2010)**

Aplicaciones comunes del RFID

El uso más común de RFID, es el de objetos con etiquetas para el control de almacenes, y tarjetas contactless, pero también se utilizan en otros muchos ámbitos.

Militar. El RFID se utilizó desde principios de la guerra en el sistema IFF, identificar si amigo o enemigo. En la actualidad están investigando un nuevo dispositivo que tenga la capacidad de comunicarse con satélites, que denominan 3G RFID.

Tracking de Paquetes de Corres: El servicio de correos y postales incorpora sistemas RFID para permitir el seguimiento de paquetes y aumentar la seguridad de los productos de los clientes.

En Sanidad: Algunos hospitales de Taiwán monitorizan sus pacientes de riesgo con pulseras con rfid para asegurarse de que se les proporciona el cuidado máximo, y también contienen información que se puede leer y modificar rápidamente por los doctores, como las dosis de los medicamentos que han tomado, alergias, etc.

Industria. Algunas empresas ofrecen partes y piezas para maquinaria pesada con RFID incorporado, donde se puede leer directamente el modelo y algunas directrices, haciendo más fácil localizar las partes necesarias. Boeing y Airbus son dos ejemplos de esto.

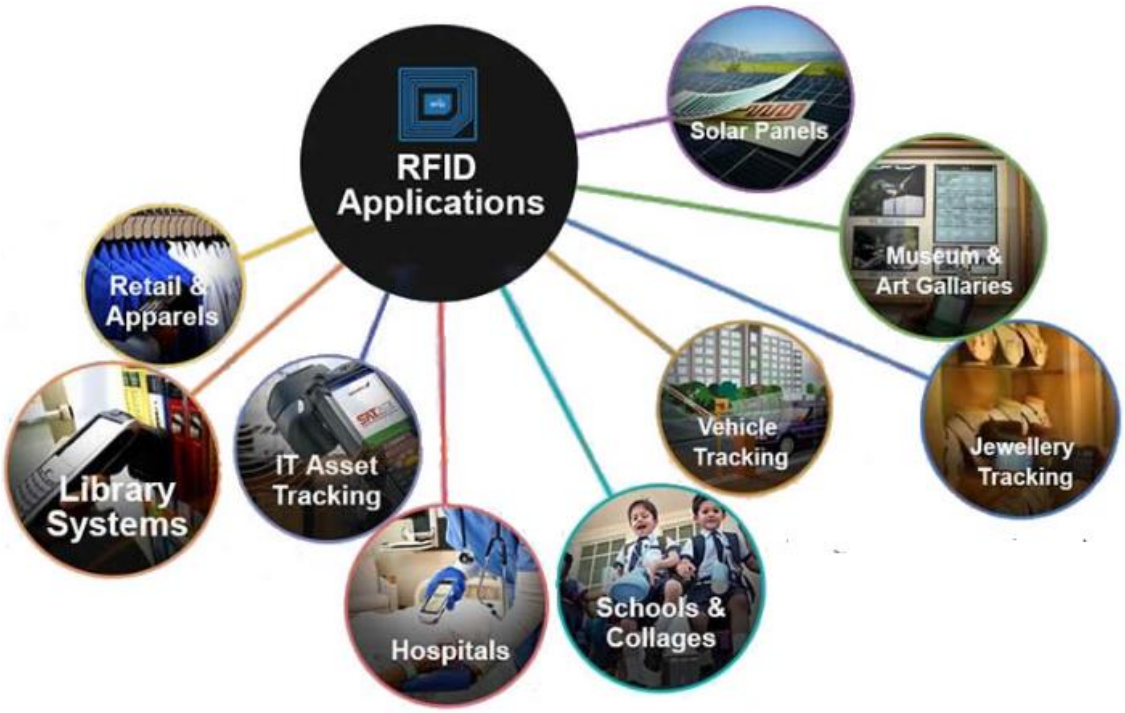


Figura 8 Aplicaciones del RFID

Implementación de Seguridad en RFID

Al igual que cualquier sistema de información y telecomunicación, los sistemas de RFID tienen el riesgo de ser potencialmente manipulado para modificar o espiar la información que contiene. Además de los ataques que puede sufrir, la seguridad y como se han de manejar los datos que contiene el sistema está controlado. En este apartado se estudiará los ataques más comunes que puede sufrir el sistema RFID, durante la recolección y transmisión de información. También se presentará y se explicará las protecciones criptográficas que protegen estos sistemas.

Para definir un sistema de telecomunicaciones como seguro y eficaz, ha de cumplir tres aspectos.

Ha de estar siempre disponible, la información del sistema ha de ser accesible para las partes del sistema que tengan autorización. Esta información ha de mantenerse ilegible siempre que no haya autorización. Si el sistema está en peligro de ser manipulado o comprometido, solo en esa ocasión ha de sobrescribirse o eliminarse.

Ha de ser confidencial, el acceso a la información únicamente ha de ser posible mediante autorización.

Mirando el contexto de uso de un sistema común de RFID, encontramos que está formada por dos grupos.

El grupo operador, que es la parte activa que se encarga de la instalación del lector y del software, además de encargarse también de la gestión de los transponders. Ya sea repartirlos además de administrar la información a la que está asociada cada tag.

El grupo de los usuarios de sistema de RFID, que suele ser o un cliente o un empleado que forma parte del sistema. Aunque los usuarios posean el transponders (ej. Una tarjeta contactless del metro), no puede influir en el uso de los datos de la tarjeta.

Para que sea seguro, ambos grupos han de poder operar, cumpliendo los tres aspectos mencionados anteriormente.

Existen varios tipos de ataques que se pueden realizar en un sistema RFID.

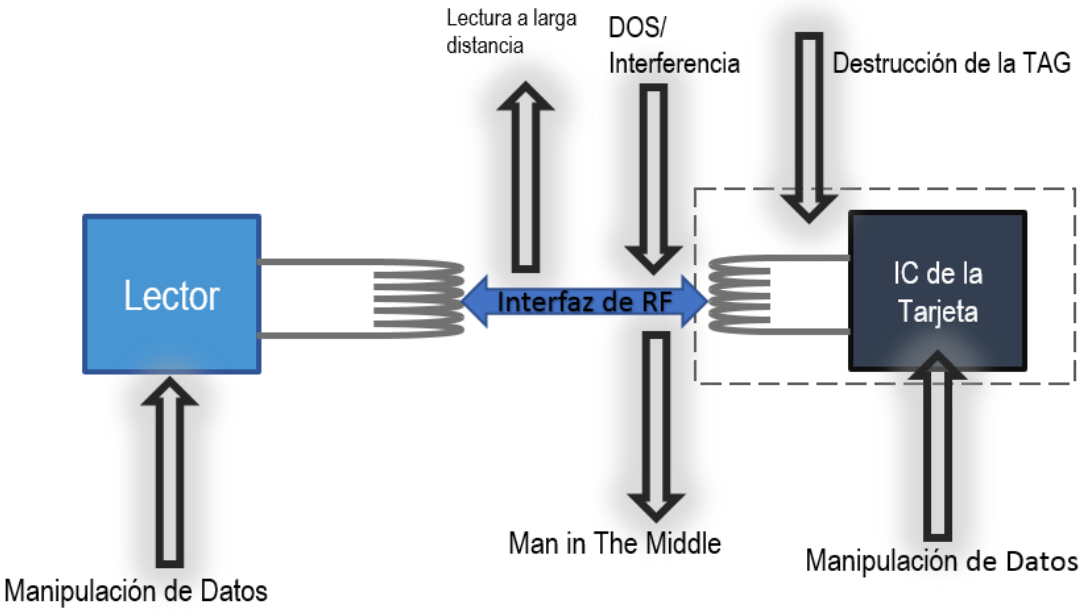


Figura 9 Esquema de ataques a un sistema RFID

Los ataques que se pueden realizar al transponder pueden ser destructivos, ya sea rompiendo la antena o el chip, también si se encuentran en un campo con la frecuencia correcta, pero si el campo tiene una fuerza superior a la que soporta la antena, esta antena se romperá debido a un sobrecalentamiento.

Otro ataque es el uso de superficies metálicas, por ejemplo, papel de aluminio de uso doméstico para cubrir (shielding) el transponder de la radiación magnética o electromagnética producida por el lector. El transponder volverá a funcionar correctamente cuando desaparezca el papel de aluminio.

La clonación de transponders es un ataque común. Si son tags sencillos, que tan solo tienen un número de serie, transmiten su información siempre que se encuentren en un campo, así que cualquier lector funcional podría leerla, y crear un clon del transponder que contenga el mismo número de serie. Puede reemplazar la PROM que contiene el número de serie, o sobrescribirla usando

switches si el atacante lee una de las tarjetas, ya que con esa tecnología no se puede saber si una tarjeta es la genuina o es clonada.

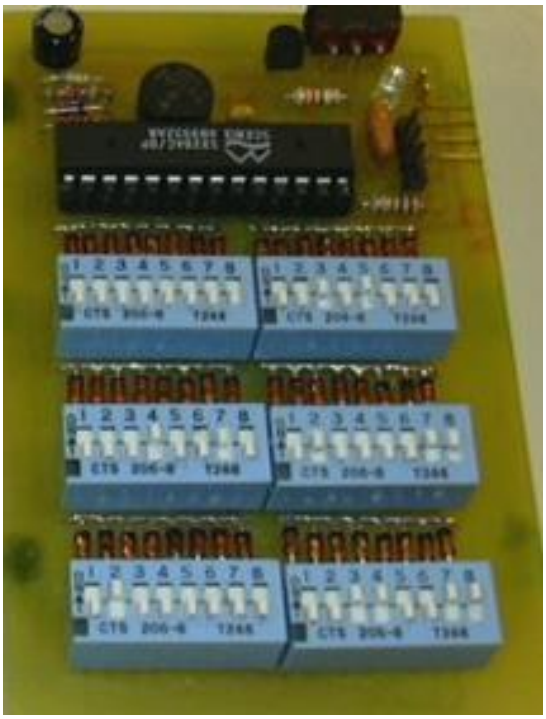


Figura 10 PCB casera de una TAG con UID "Editable"

Si son tags de los del siguiente nivel de funcionalidad, que tienen una memoria que se puede sobrescribir, es posible que esta se puede leer sin necesidad de una contraseña o clave criptográfica. En este caso, se puede igualmente leer los datos y producir copias. Sin embargo, la copia de este tipo de tarjetas nfc se puede evitar con el uso de transmisión de datos encriptados y el uso de distintos niveles de autentificación. Es por eso por lo que las aplicaciones RFID a las cuales tengan acceso un gran número de usuarios han de evitar el uso de transponders de solo-lectura o con un acceso a datos desencriptados.

Los tags pueden tener una información almacenada en ella completamente segura, pero un número de serie descubierto. En este caso se pueden realizar ataques a la privacidad del usuario. Con lectores “piratas” un atacante puede localizar al usuario de una tag de la que conozca el numero de serie, si la lleva encima. Muchas entidades estan preocupadas por la privacidad de los consumidores, que dan lugar a normas, politicas y estandares que protejan los intereses de los usuarios. Este tipo de problema se hablara en profundidad, ya que aunque se adopten normas y estandares de privacidad, se ha de suponer que un atacante no seguira estos protocolos.

También se pueden realizar ataques al lector, estos ataques son :

Un ataque común es el de interceptar las ondas de radio, que se puede hacer de manera sencilla, y por lo tanto una de las mayores amenazas de la tecnología RFID.

Se podría suponer que en parte la seguridad RFID es propia de las barreras físicas que tienen las ondas de radio, ya que se ha de estar en el rango de comunicación para poder interceptar los datos que se intercambian.

Pero debido a que la mayoría de los transponders son pasivos, estas ondas han de generar un voltaje en la antena para que el microchip funcione. Ese voltaje disminuye con la distancia, pero si suponemos como amenaza un elemento activo que sea capaz de leer esa onda con un voltaje disminuido. Es por eso que estudios que simulan este caso, muestran que el estándar de 13.56 MHz (contactless, los transponders funcionan a unos 10cm) se puede interceptar hasta en una distancia de 3m.

Este tipo de ataques se conoce como eavesdropping, y es una de las vulnerabilidades del NFC, que se estudiaran más adelante, así como las soluciones criptográficas y físicas que se toman para evitar estos ataques.

Otra variante de este ataque, donde colocando un dispositivo de transmisión, en este caso un relé, extiende la distancia de lectura. El relé está formado por dos dispositivos distintos, que simulan ondas de RFID a la misma frecuencia que el sistema de comunicación, ambas partes activas. Estas se han de situar de manera física cerca de los dispositivos legítimos de RFID, lector y tag.

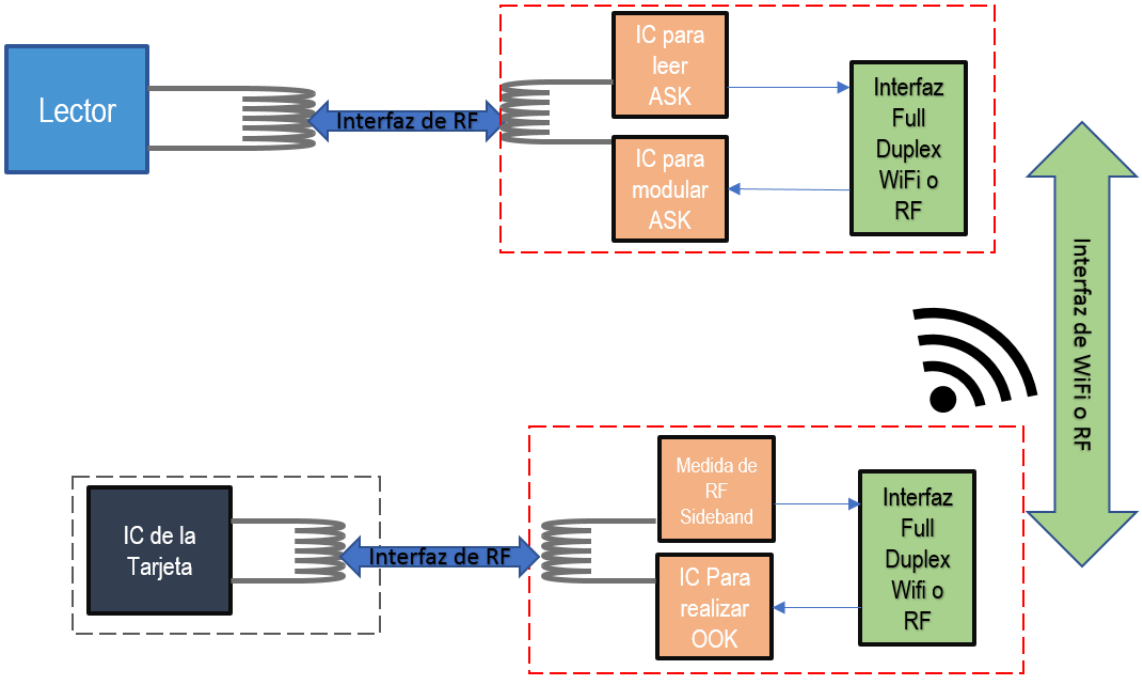


Figura 11 Esquema de un ataque Relé o MIM

La parte del relé que se encuentra cercana al lector rfid simula las ondas, y se las envía a la otra parte del relé, que se encuentra cerca del transponder. Este transponder se comunica con la parte del relé que está imitando el lector, y le comunica los datos. Es una manera de clonar la tarjeta de manera no autorizada. Estos ataques también se conocen como MIM o man-in-the-Middle.

Otro ataque sencillo es el de generar una señal que interfiera o interrumpa la comunicación. Este ataque se conoce como jamming, es un ataque de denegación de servicio en el que un dispositivo, genera ondas que bloquean la comunicación RF. Esto se hace generando ondas da la misma frecuencia, pero desfasadas. De esta forma la tarjeta se conecta, pero no recibe datos. Este ataque tiene una solución sencilla, que es con la puesta en marcha de protocolos de modulación de señal.

Los ataques de denegación de servicio (DOS), los lectores RFID modernos se comunican con varios transponders. La manera en la que selecciona uno en concreto es mediante middleware, utilizando un algoritmo anticolisión. Estos algoritmos en su forma más sencilla comprueban el número de serie. En este caso el ataque simula un tag, que, en vez de enviar un número de serie formado por 96 bits, envía 96 bits de información, simultanea siendo 1 y 0. Eso quiere decir que se generan 2^96 colisiones. Si el algoritmo que comprueba el número de serie tarda t ms en completarse, deberá de realizar la comprobación 8*10^28 veces, siendo más que improbable que entre esas colisiones se produzca la lectura de un transpondedor legítimo.

PRIVACIDAD Y NORMATIVA EN RFID

La comisión europea, estableció en 2009 una recomendación (Recomendación, una sugerencia de línea de acción sin ninguna obligación legal) una protección de los datos y de la privacidad en aplicaciones que usen identificación por radiofrecuencia.

Esta comisión estableció los requerimientos, y evaluaciones para que se cumplan estos requerimientos. A estas evaluaciones se las conoce como Evaluaciones del Impacto de la Privacidad, PIA en siglas en ingles.

Una PIA es un proceso mediante el cual un proceso consciente y sistemático se hace con el objetivo de lograr establecer los impactos a la privacidad y protección de datos de una aplicación especifica que use RFID, con el objetivo de tomar acciones que prevengan o minimicen estos impactos.

Existen los documentos de reportes de PIA, que, tras esta evaluación, se comparten con las autoridades pertinentes.

Que se realicen estas PIAs tiene varias ventajas, entre ellas:

Establecer y mantener el cumplimiento con las leyes y regulaciones en torno a privacidad y protección de datos.

Ofrecer seguridad y confianza al consumidor y al público.

Facilitar el trabajo en las fases del proyecto, y asegurando un cumplimiento de la privacidad, reduciendo los esfuerzos de diseño y de estudio de la seguridad del proyecto, ya que se sigue un protocolo.

El proceso de una PIA está diseñado para que, el encargado de la parte RFID proyecto, al que se llamara Operador, realice un estudio en el que descubra los riesgos de privacidad en relación con su aplicación, como de probables y peligrosos son estos, y los distintos pasos para encargarse de esos riesgos. Dependiendo de la cantidad de información privada, como se manipule y el tipo de aplicación, el impacto en la privacidad puede ser ínfimo o muy elevado. La PIA también establece guías y recomendaciones para lidiar con estos riesgos, y mitigar cualquier impacto en la privacidad y la protección de datos, de una manera eficaz.

Estas guías se pueden usar como base y como modelo para el desarrollo de aplicaciones industriales.

Si las tags o el sistema usan cualquier tipo de información personal o si las tags van a ser usadas por individuos (una tarjeta de acceso blanca, al tener un número de serie, también se incluye en este caso), es necesario la realización de una PIA.

Se han de delimitar los riesgos a la privacidad que puede suponer esta aplicación, y si puede ser durante la fase de desarrollo de sistema, documentar como estos peligros pueden mitigarse de manera proactiva, mediante controles u organización. La guía sirve para medir como de efectivos son estos procesos y controles. Se recomienda también que estas estrategias se apliquen en el diseño del sistema, y no unirlas una vez se finalice. En el anexo de este TFG se encuentran los distintos objetivos para cumplir la normativa de privacidad.

Estos objetivos, en resumen, buscan la calidad y la legitimidad de todos los datos guardados en el sistema RFID, y que cualquier usuario de este sistema tenga la seguridad que se no se vulneran sus derechos en cuanto a materia de privacidad, según lo establece el Directivo de Privacidad de la UE.

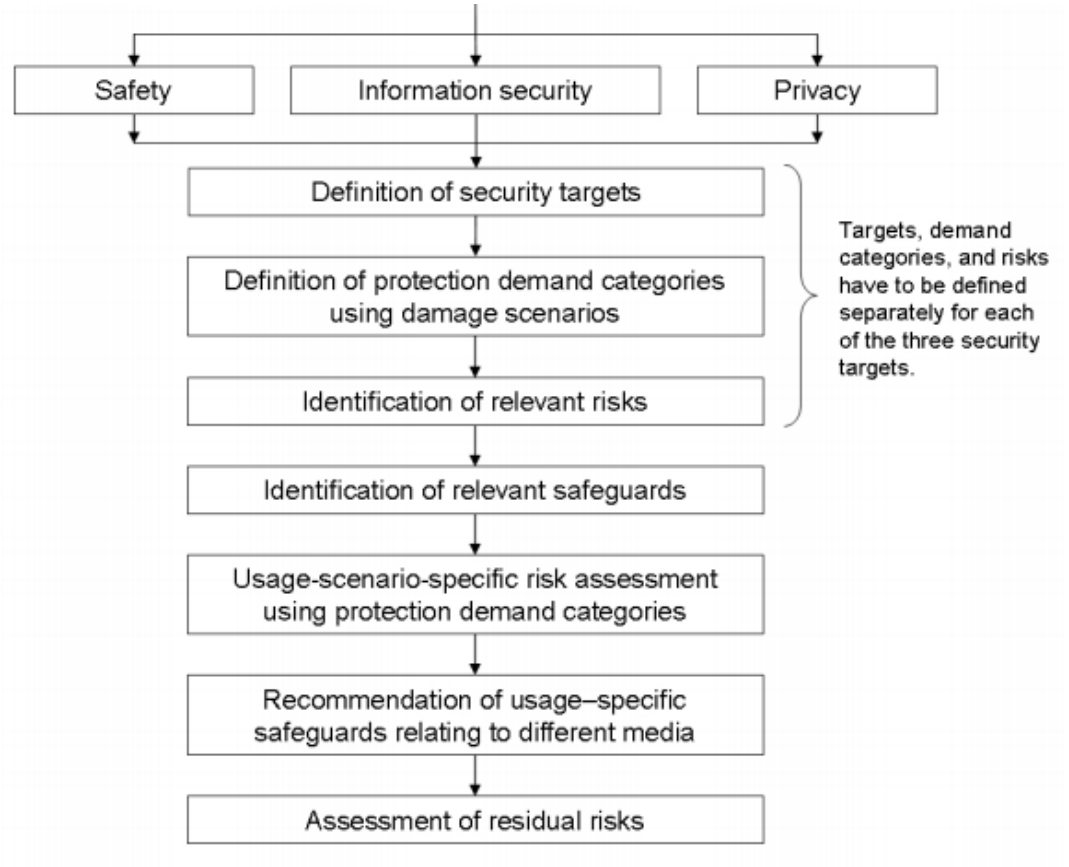


Figura 12 Diagrama de evaluación de riesgos según la PIA

Los sistemas RFID se usan en aplicaciones de alta seguridad, como sistemas de pago, tickets o sistemas de acceso. Estos sistemas necesitan medidas de seguridad para protegerse de posibles ataques, en los que la gente intenta engañar a el sistema RFID para conseguir acceso no autorizado u obtener servicios o dinero. Como ya se ha explicado, un sistema RFID seguro ha de tener defensas contra ataques que busquen clonar las tarjetas u obtener información durante comunicaciones.

(Arnaud, 2011)

Además, al elegir un sistema se ha de tener en cuenta el tipo de aplicación. Por ejemplo, un sistema RFID de etiquetas para asegurarse que los mecánicos de un taller no abandonen las herramientas dentro de un avión, se añadiría un sobrecoste si se añaden procedimientos de seguridad. Pero viceversa, si por ejemplo un sistema de tickets de metro no tiene un sistema de seguridad, puede ser una equivocación muy costosa si se puede acceder a los servicios que estos tickets dan, sin autorización.

CC EAL CERTIFICACTION

CC, o Common Criteria, es un estándar internacional de la Evaluación de Información y Seguridad tecnológica, y está basado en la norma ISO 15408.

Este funciona a partir de un nivel de seguridad, denominado EAL, Nivel de Seguro de Evaluación. Este nivel se le asigna a un producto tecnológico después de una evaluación desarrollada por common Criteria.

La mayoría de las aplicaciones de NFC, que tienen datos personales o algún tipo de método de pago o de banking, se encuentran en EAL5 o EAL6, o alguna de sus variantes intermedias.

Figura 13 Logo de CC

La seguridad no se puede definir como un valor absoluto, por lo que el ranking de EAL depende del producto, el campo en el que se utilizara y su propósito.

No implica que un producto tenga mejor seguridad, pero mide el número de test a los que se ha sometido, basado en los usos y necesidades de ese producto. Por eso todos los productos tienen un documento llamado objetivo de seguridad. En este se explica el propósito, sus funciones, hardware y software y las funcionalidades de este producto valorados en una evaluación.

Esta evaluación tiene en cuenta las amenazas posibles, supuesto y requerimientos funcionales. En función de esta evaluación, y con unos requerimientos mínimos de conformidad, se consigue la valoración EAL. No se pueden comparar EAL para distintos productos, sin tener en cuenta el contexto de la evaluación. Los niveles 5 y 6, presentados en el Caso de estudio de este TFG, implican que la ingeniería de seguridad de la aplicación tiene un desarrollo riguroso y con técnicas especializadas, que puede proteger objetivos de alto valor a ataques significativos.

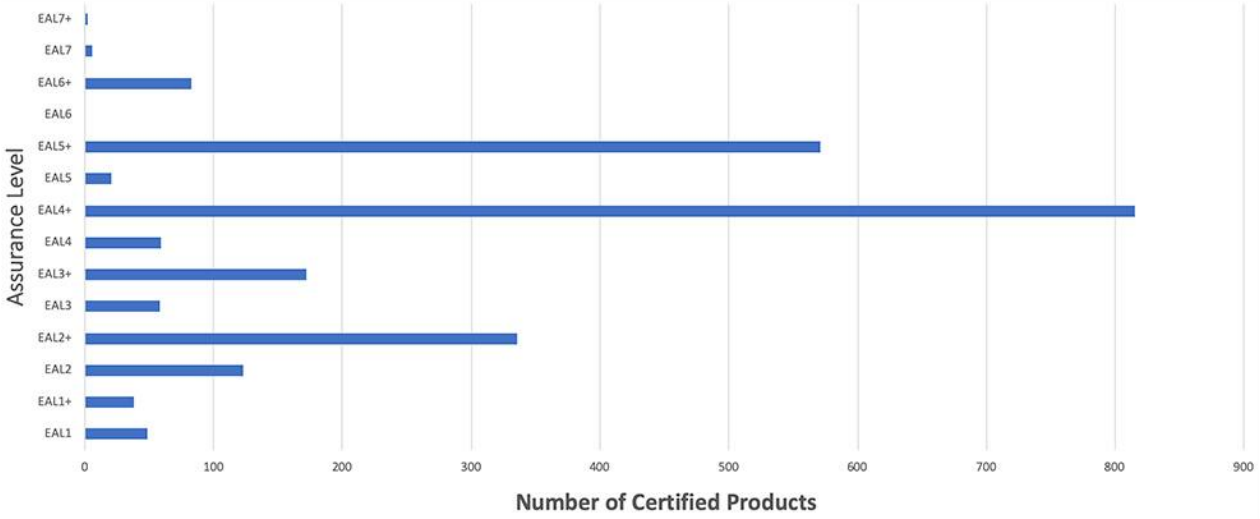


Figura 14 Numero de productos certificados por CC y sus valores

NFC, La tecnología de comunicación de campo cercano

Introducción

El desarrollo de estándares en responsabilidad de un comité formado por varias instituciones encargadas de la estandarización. ISO, de las siglas en ingles Organización intencional para la Estandarización, es unión global de instituciones de distintos países, como por ejemplo la UNE española o DIN alemana, entre otras. Debido a que este proyecto se centra en una tecnología en concreto, el NFC, los estándares que rodean esta tecnología serán los que estudiaremos.

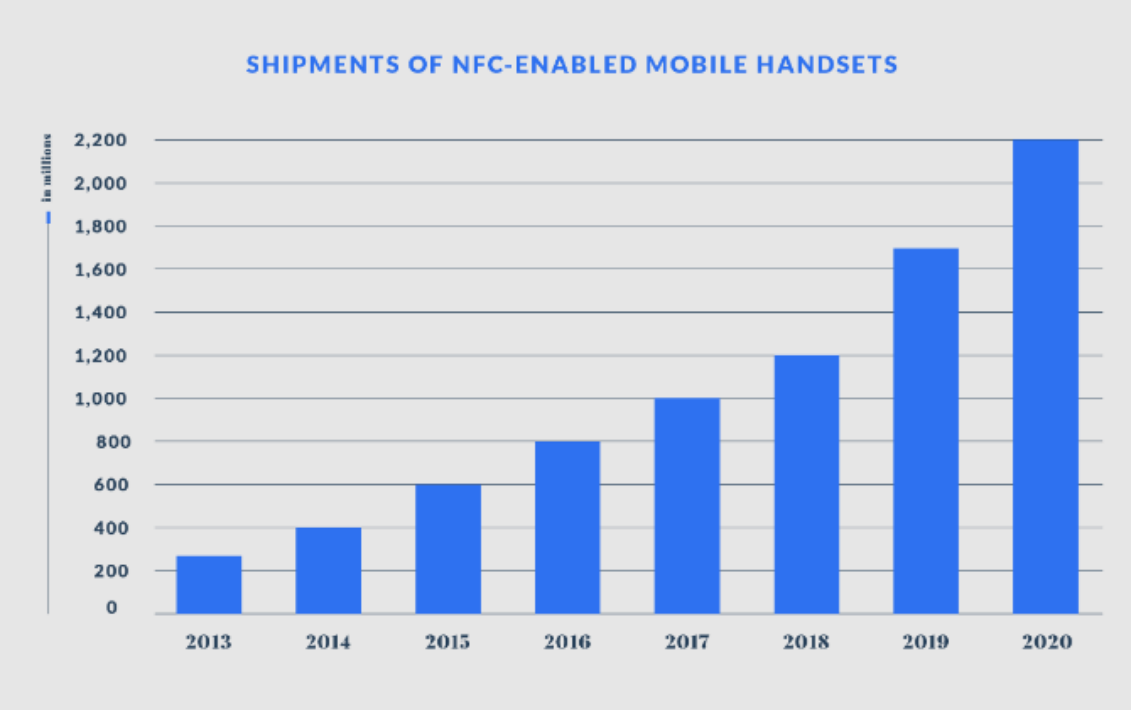


Figura 15 Grafico del número de teléfonos móviles con tecnología NFC incorporada

Esta tecnología se utiliza tanto para el intercambio de datos entre dispositivos “Smart”, como para tarjetas de crédito del tipo contact-less como para la emulación de dichas tarjetas en smartphones.

.

NFC

El NFC, Near Field Comunication, comunicación de campo cercano, es una tecnología de comunicación basada en la tecnología RFID, y originada como plataforma para la transmisión de datos entre teléfonos móviles.

Los expertos predicen una explosión de la interconectividad de las “cosas”. Cuantas más cosas estén conectadas, mayores serán las preocupaciones en cuanto a la privacidad de datos y la seguridad. La tecnología NFC está enfocada a solucionar esas preocupaciones. Por diseño, el NFC tiene un campo de operación reducido a unos pocos centímetros, lo que evita que se puedan filtrar datos. En contraste con el Wifi, que tiene un flujo de información constante, el NFC solo envía datos cuando se produce esta conexión. Junto con protocolos de seguridad y criptografía, el NFC asegura que solo se produce intercambio de datos de manera intencional.

Figura 16 NFC dentro del IoT

Si se compara con otras tecnologías de intercambio de información sin cables, todas tienen sus beneficios y desventajas. Entre estas tecnologías, el NFC tiene 2 ventajas: Máxima privacidad, operaciones sin apenas gasto de energía y un coste muy reducido, añadir un tag nfc a un sistema embebido establece

En este capítulo se hablará de los estándares que definen esta tecnología 14443 y 18092. En otro capítulo se hablarán de los estándares ISO que se aplica a la criptografía del NFC

Los dispositivos establecen comunicación entre si a los pocos centímetros, y ofrece varios usos que otras tecnologías no pueden ofrecer en cuanto a lo que transmisión de datos se refiere. La tecnología NFC funciona usando inducción de campo cercano para comunicarse con dispositivos que se encuentren en el rango de uso.

El NFC funciona a la frecuencia de 13.56MHz, y ofrece velocidades de transmisión de hasta 424kbps. El principal atractivo de esta tecnología es que la información se intercambia de manera transparente, segura y eficiente.

Esta tecnología viene integrada en la mayoría de los móviles actuales, y permite a las personas integrar cualquier tipo de tarjeta o tag NFC, en sus teléfonos móviles.

En la actualidad, Apple ha anunciado que en 2021 que serás capaz de añadir una llave de coche digital, y que “podrás usarla para abrir el coche, usando el NFC de tu IPhone o Apple Watch” “Cuando unas tu llave al teléfono, y un coche que soporte estas llaves digitales, solamente tendrás que acercar el dispositivo Apple al lector NFC incluido en la puerta del coche” “Como características añadidas, Apple anuncia que será posible compartir la llave del coche a través de la aplicación, podrás limitar la velocidad y la aceleración del usuario de ella llave, y quitarle el acceso en cualquier momento. Funcionan sin conexión a Internet” *En anexos se encuentra el hipervínculo a la noticia*

En el apartado RFID se ha explicado por encima el concepto de modulación de señal para comunicarse.

El sistema de modulación que se usa en NFC, dependiendo de la ISO 14443 que se utilice, puede ser distinto.

NFC Technical Standards Specifications of the Air Interface					
NFC-Forum Standard	**Polling / Listening**	**Coding**	**Modulation**	**Data Rate**	**Carrier frequency**
NFC-A	Polling	Modified Miller	ASK 100%	106 kb/s	13.56 MHz
	Listening	Manchester	Load modulation (ASK)	106 kb/s	13.56 MHz +- 848 kHz subcarrier
NFC-B	Polling	NRZ-L	ASK 10%	106 kb/s	13.56 MHz
	Listening	NRZ-L	Load modulation (BPSK)	106 kb/s	13.56 MHz +- 848 kHz subcarrier
NFC-F	Polling	Manchester	ASK 10%	212 / 424 kb/s	13.56 MHz
	Listening	Manchester	Load modulation (ASK)	212 / 424 kb/s	13.56 MHz (without subcarrier)

Figura 17 Distintos estandares de modulacion y codificacion de señal NFC

14443-A utiliza modificación on/off, que puede ser tanto ASK, que modula la amplitud o FSK que modula la frecuencia.

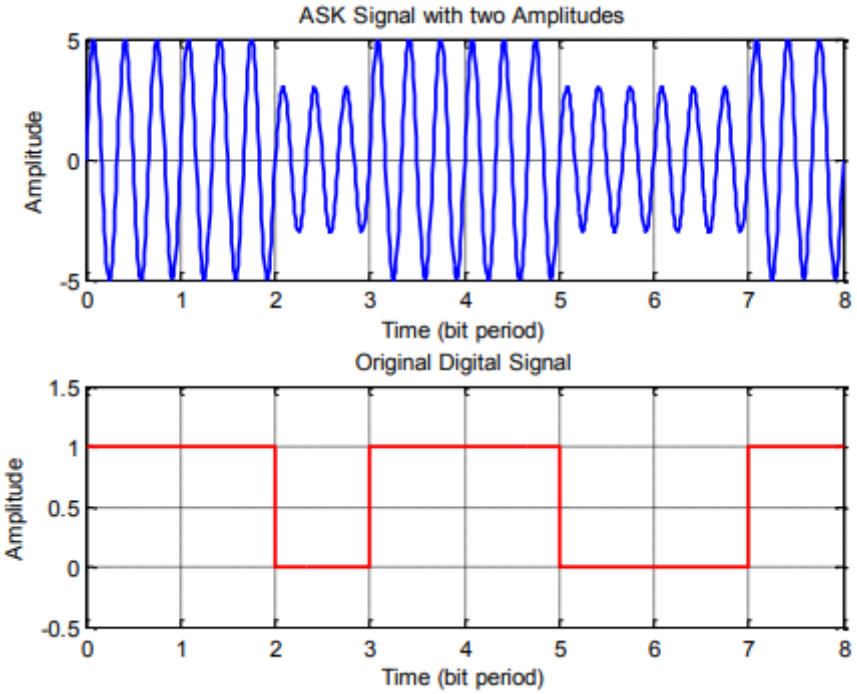


Figura 18 ASK, modulacion por amplitud

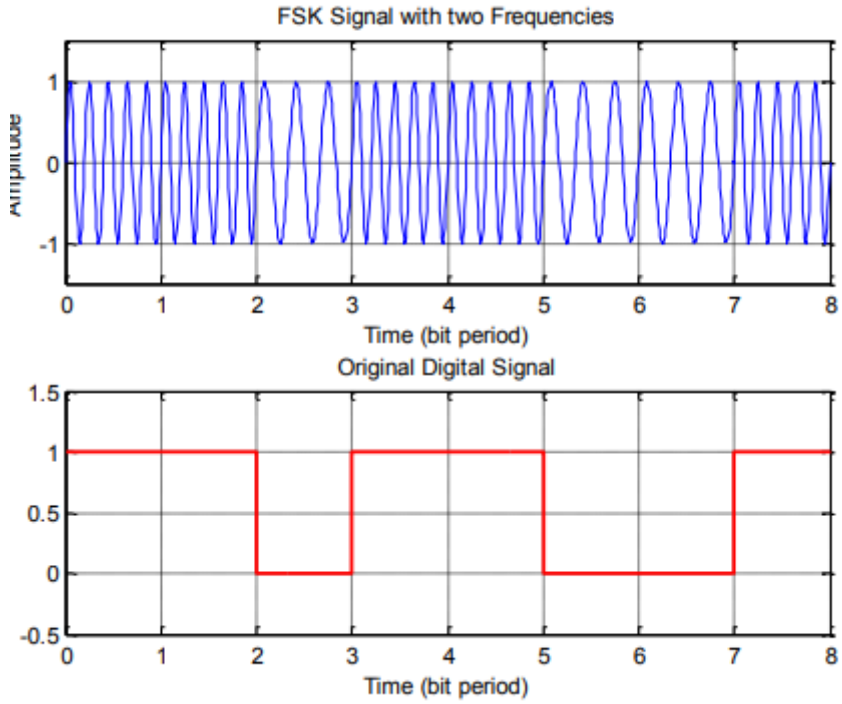


Figura 19 FSK, Modulacion por Frecuencia

La iso 14443-B utiliza BPSK, modulación por desplazamiento de fase, para la lectura

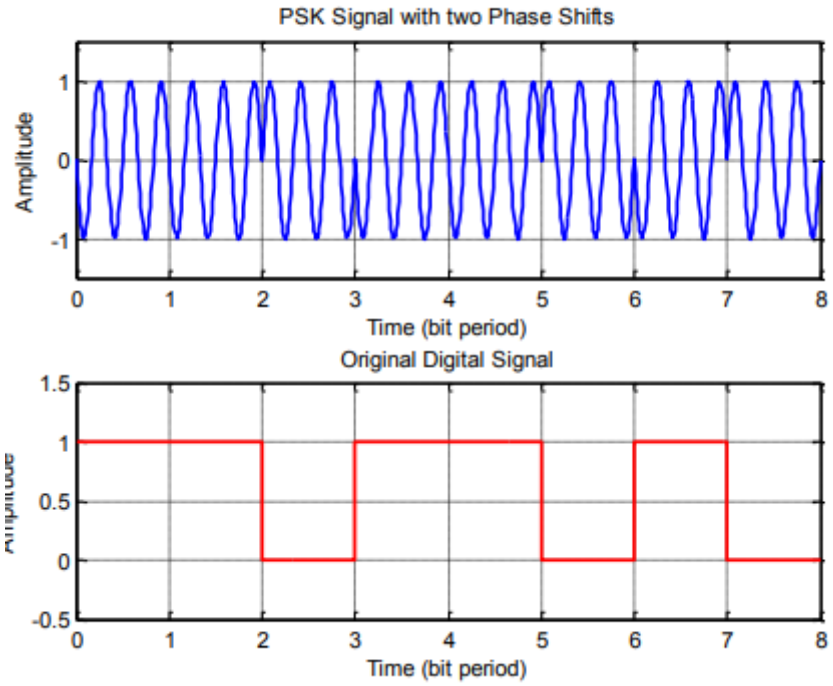


Figura 20 BPSK Modulacion por desplazamiento de Fase

Además, NFC A y B toman distintas maneras de codificar cuando una señal envia un bit 0 y cuando envia un bit 1. Estos pueden ser

NRZ-L. Un estado alto indica un bit 1, y un estado 0 indica un 0.

Manchester. La duración de cada bit se divide en dos estados, primero alto luego bajo para indicar un 1, y primero alto después bajo para indicar un 0.

Modified Miller. La duración de cada bit se divide en dos. Si se envía un pulso bajo tras la primera mitad del bit, es un 1. Si se envía un pulso bajo nada más comienza el bit, es un 0. Si un 0 sigue a un 1, no se envía pulso.

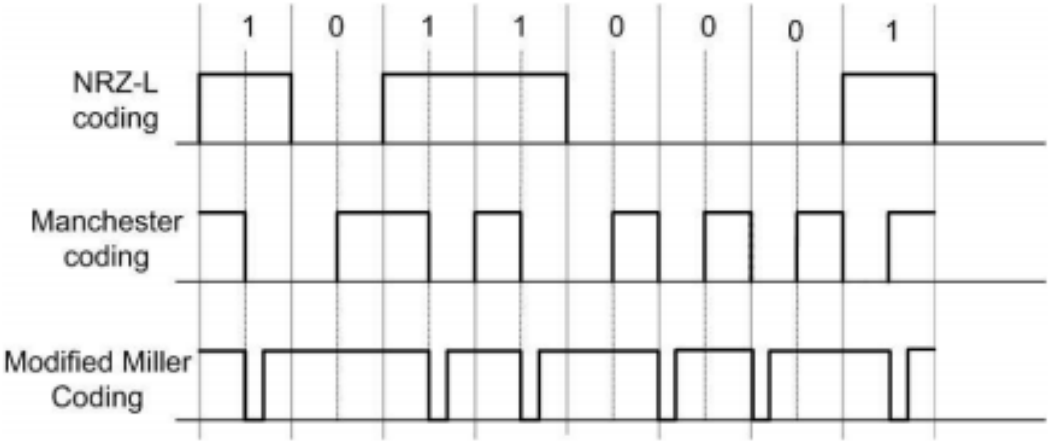


Figura 21 Distintas codificaciones de señal para bits lógicos en NFC

Todos los dispositivos activos, de la tecnología NFC pueden actuar como lectores, siendo los activos, o como “leídos” siendo pasivos en la comunicación y modulando señal, sin usar la energía de la batería. En contraposición a lo explicado hasta ahora en los conceptos de RFID, que diferenciábamos los dispositivos en activos y en pasivos, ahora los dispositivos activos (móvil, smartwatch) presentan dos modos, activo y pasivo. En el modo pasivo, este actúa como un tag, y no genera ningún campo RF, solo lo modula.

. Todos los dispositivos nfc están en modo activo, siempre, hasta que alguna aplicación les pida estar en modo pasivo.

El proceso de comunicación es el siguiente.

El elemento activo inicializa la comunicación, y busca detectar un único dispositivo. Antes de comunicarse se asegura de que no haya ningún otro campo RF para evitar colisiones y no molestar otras comunicaciones.

Según la respuesta que, de el otro dispositivo, este pasara a modo activo o pasivo. Se elegirá un tipo de comunicación por parte del dispositivo que la inicializó.

Después se seguirá el Protocolo de Intercambio de Datos (DEP), y tras finalizar la comunicación, el protocolo de terminación (RLS)

Tipos de comunicación

El NFC tiene 3 tipos o modos de comunicación.

Lectura/Escritura. Cuando un dispositivo lee datos de tags u otros dispositivos con NFC, y actúa en esta información. Un ejemplo de esto es el uso de un

teléfono móvil como lector, al acercarlo a un tag NFC, este lee la información del tag. Este tag puede contener información o acciones sencillas para el dispositivo, como activar un despertador, conectarse al wifi. Este dispositivo también puede escribir en el tag, si conoce la clave de escritura.

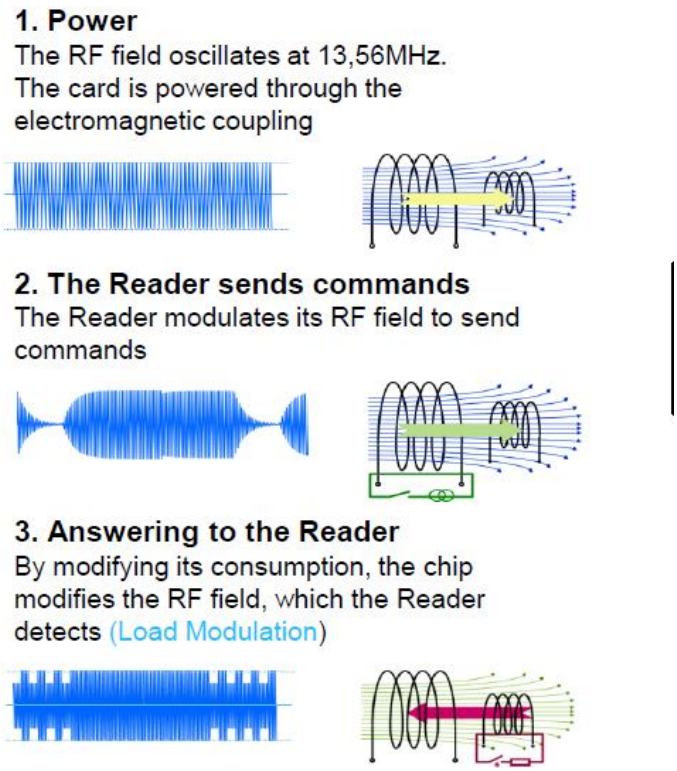


Figura 22 Funcionamiento de Lectura Escritura

Emulación de Tag. Un dispositivo, que contenga un elemento seguro (explicado aquí), puede simular se una tarjeta contactless. En el caso de un teléfono móvil, la tarjeta Sim actúa de SAM, lo que permite usar el dispositivo emulando una tarjeta para pagos, tickets de metro, acceso a diversos sitios, etc. En este caso el dispositivo que emula la tag, lo hace de forma pasiva.

Peer-to-Peer. Un dispositivo Smart puede comunicarse con otro, para cambiarse datos, con la misma seguridad que el modo lectura escritura. Ambos dispositivos son capaces de comunicarse. Un dispositivo actúa como tag, pero este puede pasar a ser el “lector” si así lo requiere la comunicación. El dispositivo que inicia la comunicación gasta menos energía que en el modo lectura/escritura ya que el otro dispositivo utiliza su propia fuente de energía.

(N.a, 2011)

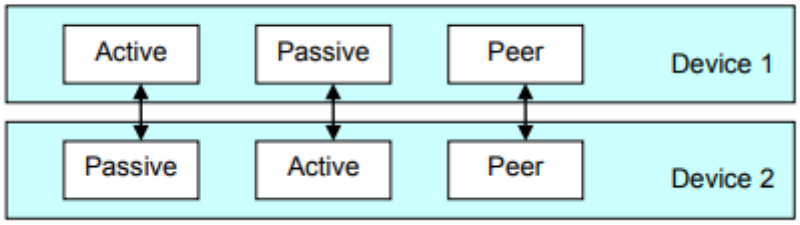


Figura 23 Distintos tipos de comunicación entre dispositivos NFC

Diseño y desarrollo de un sistema NFC

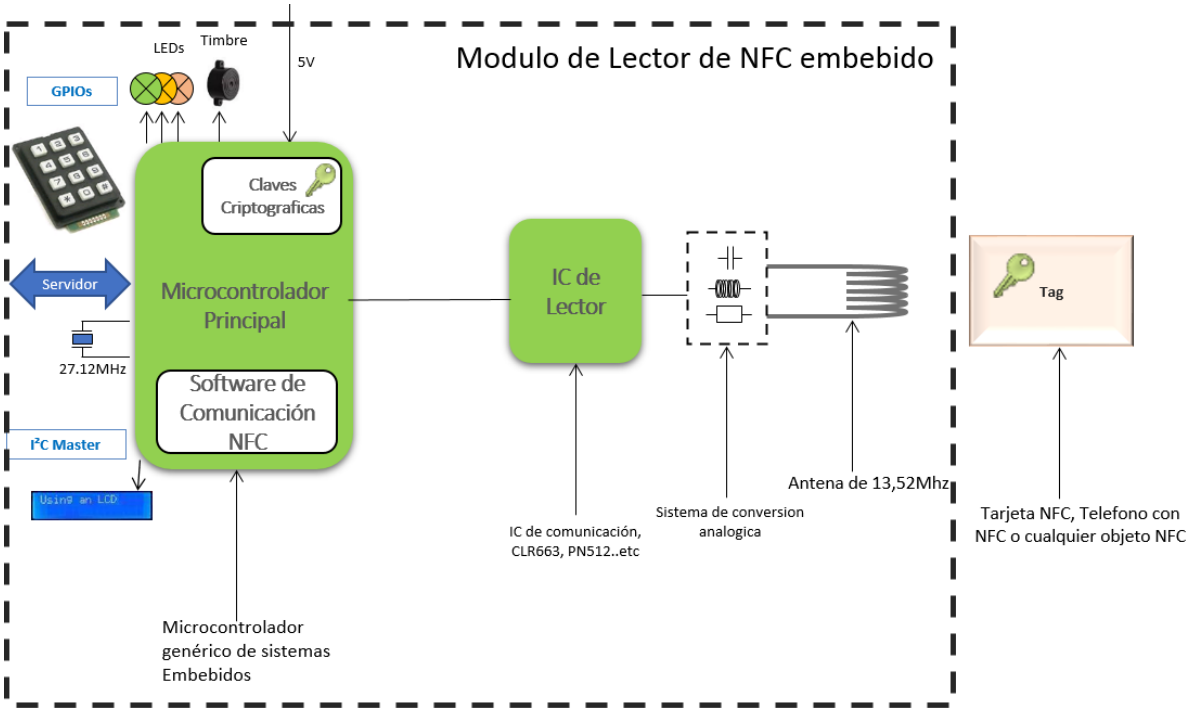


Figura 24 Arquitectura basica de un lector NFC

El proceso básico de diseño de un sistema NFC es el siguiente:

- Seleccionar un Circuito integrado que controle la comunicación RF
- Selección de un sistema. Lo que será el “cerebro” de nuestro sistema.
- Selección de la arquitectura de seguridad.
- Diseño final, acabado.

Cuando seleccionemos un circuito integrado para controlar el sistema del lector, se ha de tener en cuenta que: El microcontrolador elegido pueda soportar [ISO 14443-A y B] :

Los requerimientos de P2P y R&W establecidos en NFC Forum.

Una interfaz de comunicación con el Host, SPI, UART, I2C.

Tenga un interfaz analógico que permita la modulación y desmodulación de la señal de la antena.

En el caso práctico se ha optado por el CLR663

Como microcontrolador, uno que permita la comunicación con interfaces externas, (Serie, USB o Ethernet). Dependiendo de la aplicación, y con capacidad de procesamiento que permita realizar las operaciones criptográficas necesarias.

La mayoría de los microcontroladores no están diseñador para poder mantener de manera segura llaves criptográficas (Algunos tienen una EEPROM, pero no es lo más seguro)

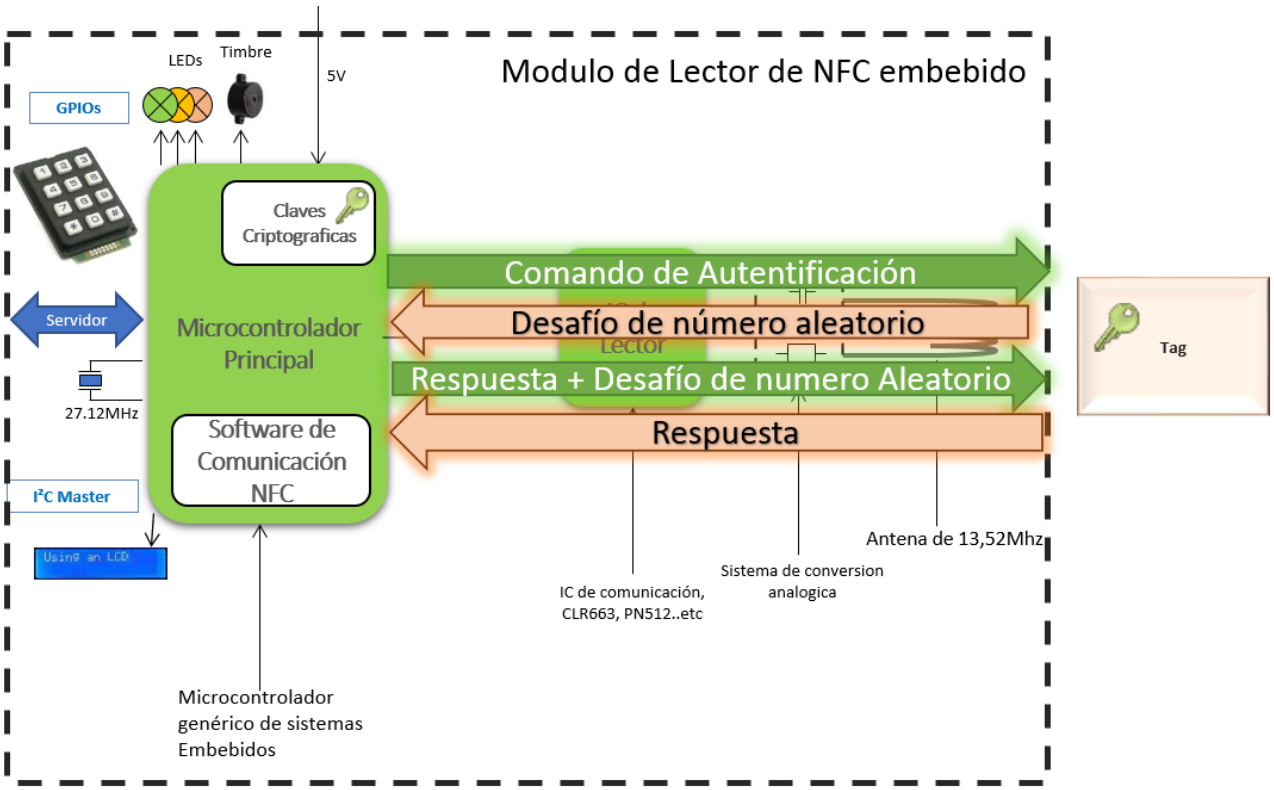


Figura 25 Diagrama de comunicación NFC sin SAM

Se puede añadir una interfaz SAM, que tiene hardware embebido que permite realizar las operaciones criptográficas de manera eficiente, además de almacenar las claves, lo que lo convierte en un sistema seguro.

Como se generan tarjetas

En todos los sistemas de criptografía hemos asumido que ambos dispositivos tienen una clave privada, y en el caso AES que la comparten. ¿Pero cómo se ha introducido esa key de manera segura en el dispositivo? Esta distribución es un problema y supone una parte crucial en la seguridad de un sistema.

En muchas ocasiones estas llaves privadas de los transponders se generan y programan durante la manufacturación. Aun así, en muchas ocasiones las aplicaciones o las estaciones en las que se utilizan estos transponders han de poder aprender de una manera segur en caso de que haya que añadir, por ejemplo, una nueva tarjeta para acceder.

La transmisión de estas a través de una interfaz RF no es seguro, y la opción que tenemos es mediante sistemas de llave publica RSA. Aun así, solo las tarjetas contactless de última tecnología y para funciones de pago o de muy alta seguridad las utilizan, ya que generan un coste adicional en el poder computacional de los transponders.

Una solución sencilla es utilizar la misma clave, o conjunto de ellas, para todos los dispositivos del sistema, y entonces todos los lectores conocen esta clave desde su fabricación. Este tipo de seguridad se conoce por SPOF o Único Punto de Error. Si un atacante lograra la llave, todos los dispositivos estarían comprometidos. Es por eso por lo que en los protocolos actuales se recomienda el generar una única clave secreta para cada transponder y estación.

Otros métodos permiten que las llaves se generen y almacenen en una base de datos y servidores seguros de la empresa manufactura dora. Así, estas claves se podrían programar en un sistema en el que ningún atacante puede monitorizar el proceso. Así se podría añadir al sistema nuevos transponders después de la instalación. Aunque sea más seguro, sigue presentando un SPOF que es una base de datos conteniendo todas las claves secretas.

En pasaportes electrónicos y DNI 3.0, la clave secreta es derivada directamente de los números impresos en el documento. (Basic Access Control) BAC. Estas llaves no son secretas y pueden ser obtenidas por cualquier persona con acceso físico al documento.

Ataques y amenazas de la tecnología NFC

Una de las razones por la que el NFC se ha estandarizado es por su seguridad. La comunicación ha de realizarla un usuario debido a que ha de comunicarse por una corta distancia. Pero esta distancia no implica que este sistema es seguro solo por esta restricción física, ya que pueden existir varias amenazas que afecten a la comunicación NFC.

Una de las mayores amenazas es que el dispositivo que comunica, el tag, haya sido manipulado de manera física, ya sea quitando el tag del dispositivo al que está unida o cubriéndola con papel de aluminio, entre otras.

Dependiendo del tipo de etiqueta NFC que se use, y si la información que contiene se puede sobrescribir si no se almacenan bien los códigos necesarios para la sobreescritura. Estos códigos o claves establecen que dispositivos son los que conocen la clave para modificar el contenido del tag, y en ocasiones también para comunicarse. Para evitar errores durante la comunicación, el NFC utiliza CRC o check de redundancia cíclica, que permite ver si los datos que se han comunicado se han modificado o corrompido durante el camino. Los ataques más comunes que puede sufrir el NFC son las siguientes. (Dennis Giese, 2018)

Eavesdropping. Que es uno de los ataques más comunes realizados todo tipo de comunicaciones sin cable, NFC incluido. Un atacante usara una antena para recibir las señales transmitidas, y si tiene el conocimiento necesario y el sistema no es seguro, será capaz de decodificar los datos. La solución para el eavesdropping es el establecimiento de un canal seguro usando un protocolo de clave publica basado en Diffe-Hellman, que da integridad y confidencialidad a la transmisión.

Corrupción de datos. Un atacante puede intentar destruir o corromper los datos que se transmiten vía NFC. Puede intentar sellar la comunicación y el resultado sería que el servicio de comunicación no estaría disponible. Este método se conoce como jamming y se logra transmitiendo frecuencias que se correspondan al mismo espectro que los datos, que se puede hacer si el atacante conoce el método de modulación y de codificación de datos, y este ataque es sencillo si el objetivo del atacante no es el de modificar datos si no el de denegar el servicio. (DOS). Este tipo de ataque se puede prever, pero también es fácil de detectar ya que la energía necesaria para hacer una señal de jamming es más elevada que la necesitada para la comunicación NFC.

También en aplicaciones militares que usan RFID, se utiliza un desplazamiento de frecuencia (Spread Spectrum), que se emplea en la transmisión de datos, en varias frecuencias para prever tanto interferencias como intercepción de señales.

Modificación de datos, a diferencia de la corrupción, en ocasiones el atacante busca modificar y manipular los datos. Puede hacerlo durante la comunicación, debido a que en la modulación de datos que se utiliza, Modified Miller Code y Manchester coding.

El atacante puede cambiar el valor de un bit por el que desee. El atacante tiene que transmitir una señal que se superponga con la señal original, y a su vez transmitir una nueva señal. La solución para este tipo de ataques es que la parte activa busque detectar otros campos RF durante la comunicación. Si esto es así, y se detecta una amenaza la comunicación se detiene. Otra solución que requiere de un procesador capaz de realizar funciones criptográficas es el establecer un canal seguro mediante algoritmos del tipo. Diffie-Hellman. También se pueden insertar datos, en esta ocasión el atacante enviara datos durante la comunicación. Si sabe cuándo un dispositivo responde y tarda tiempo en hacerlo, existe un delay con la respuesta, el atacante podría aprovechar ese delay para enviar su mensaje. Un canal seguro, es una solución a este problema.

Man in the middle, es un tipo de ataque en el que los dispositivos no se están comunicando entre ellos, si no que están enviado y recibiendo datos de un atacante, que simula la conexión. En este caso el atacante buscara cortar la comunicación entre ellos, y situarse para comunicarse tanto con receptor como con transmisor. La solución a este problema es estableciendo un canal seguro de comunicación, que no permita que un elemento que no sepa la clave privada se comunique.

(Breitfuß)

NDEF, Como son los mensajes NFC.

Los mensajes han de seguir un mismo protocolo o estructura para que se puedan entender. Este protocolo se denomina NDEF, cuyas siglas significan formato de intercambio de datos de NFC.

Cada “mensaje” se transmite dividido en varios bloques llamados registros, que están formador por un encabezado, y el “contenido”, payload en inglés, que es la información que se busca transmitir.

(Motlagh, 2012)

En este encabezado se envían:

FLAGS, 1 byte de información, que a su vez está formado por 5 flags, de un bit cada una y un formato, 3 bits que indican la estructura del mensaje y que norma de “compresión” hay que seguir para interpretarlo. (NFC RTD, NFC Forum External, URI o Media)

Type Length, algunos tipos o normas necesitan distintas longitudes.

Payload Lenght, expecifica la longitud del payload.

ID Lenght. En caso de que el mensaje NDEF requiera de identificación. Solo se escribe si la flag IL esta activada.

Payload Type, especifica la norma que se usa.

Payload ID. Especifica el ID del registro NDEF, en caso de ser necesario.

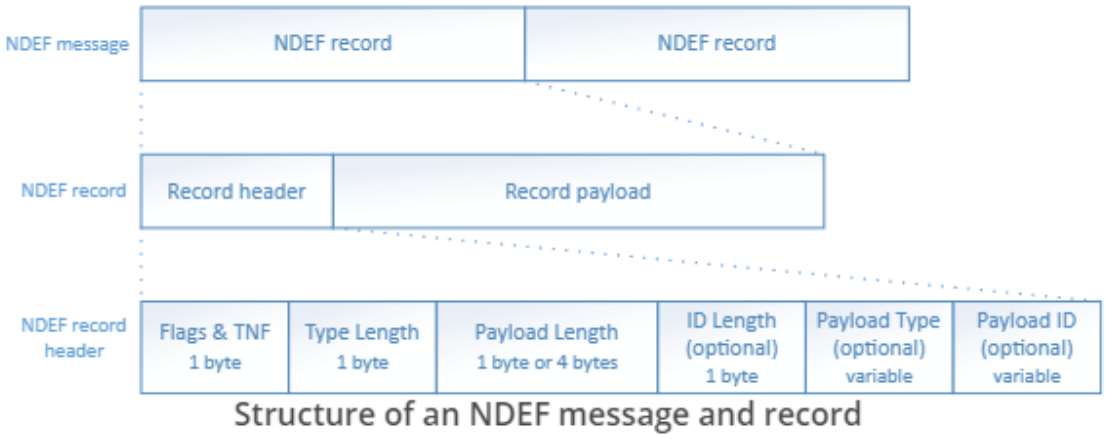


Figura 26 Estructura del bloque de datos NDEF

Aplicaciones del NFC.

El uso de NFC ha variado mucho en la última década, y podemos encontrar NFC en varios productos que usamos a menudo.

Electrodomésticos: Electrodomésticos, como por ejemplo las lavadoras, son productos donde los valores estéticos son importantes. El tamaño esta estandarizado por lo que los márgenes de mejora son moderados. Los usuarios quieren características especiales, funcionalidad y fácil manejo y operación. El NFC es un sistema económico con el cual el manufacturador puede crear una aplicación en un dispositivo móvil con NFC incorporado, donde operar la lavadora sea sencillo e intuitivo, con muchas funcionalidades. Extenderlo a una gama de productos creando un propio ecosistema de electrodomésticos.

“Wereables” : Los brazaletes o otros dispositivos wereables son dispositivos que permiten al usuario marcarse metas deportivas y de fitness y medir el progreso hacia esas metas. Por definición estos dispositivos han de ser pequeños y recargables, así el tamaño de pantalla ha de ser pequeño, por lo que añadir un tag nfc al dispositivo da lugar a una mejor experiencia de usuario, al poder marcar esas metas deportivas en su teléfono. Además esta corta distancia protege los datos privados del usuario

Contador Inteligente.

Los contadores inteligentes se están volviendo más comunes, ya que a las compañías no utilizan los contadores digitales o mecánicos, que actualmente solo sirven para el usuario. Si se añade un chip embebido NFC al diseño del contador, se puede prescindir de la parte mecánica del lector y utilizar la comunicación nfc para expandir el display del medidor a un dispositivo con nfc como sería un teléfono móvil, y además se pueden colocar gráficas y perfiles de usuario, gasto mensual etc.

Control de termostato.

Las interfaces de control de termostato en zonas comerciales u oficinas. Lo que unos rangos de temperatura determinados para unas personas están bien, para otros son demasiado cálidos o fríos para el confort. En muchas ocasiones los termostatos son confusos, difíciles de ver, difíciles de manipular. Con una interfaz embebida NFC, estos controles pasan a ser sencillos de usar, al conectar un teléfono móvil con tecnología NFC, el display pasa a ser una interfaz gráfica que permiten ajustes, control diario, y otras características avanzadas.

Acceso a dispositivos Electrónicos.

Colocar un chip embebido en los productos electrónicos proporciona un acceso controlado. Usuarios con un tag o un teléfono móvil con las credenciales apropiadas pueden acceder a usar dispositivos, que otras personas sin dichas credenciales no pueden usar.

Esto recude el coste de otras medidas de control externas, añade facilidad de uso, inclusive si se añade una interfaz gráfica para el uso, lo que simplifica a su vez el diseño, ya que reduce el número de botones y dispositivos externos, y aumenta la seguridad.

Mantenimiento autónomo.

Cuando un sistema tiene un sistema embebido NFC, el sistema puede usar la memoria no volátil del tag para almacenar números de errores, modelos y números de serie e información de la garantía, y links a guías del producto o datasheets. Además, para acceder a esta información no es necesario que el sistema tenga conexión, ya que este sistema puede actuar como NFC pasivo.

Ahorra dinero, los consumidores tienen más facilidad para pedir recambios. Se reduce el número de documentación en papel necesaria, y aplicaciones en la web pueden dar información específica en función del número de error que haya almacenado en el tag.

Wifi y Bluetooth pairing: Con el sistema NFC, los consumidores pueden realizar la acción de gestión de códigos bluetooth y contraseñas Wifi, eliminando los pasos de buscar dispositivo, añadirlo manualmente y colocar la clave. Ofrece mayor seguridad que el mantener la clave escrita en un papel o los protocolos Wireless de larga distancia para compartir claves.

Pagos mediante móvil, para tickets, taxis, y en POS contactless.

Sistemas de acceso, firmas electrónicas, acceso a seguro a PC o maquinaria, almacenamiento de tags nfc de acceso en teléfonos móviles.

Acceso a edificios seguro, donde puedes extender una tarjeta NFC a los huéspedes y debido a la seguridad propia de esta tecnología, saber que es imposible que te hayan clonado la tarjeta y en caso de robo o extravío, se puede eliminar del sistema.

Transmisión de datos entre distintas unidades con NFC (peer-to-peer P2P), como por ejemplo tarjetas de negocios electrónicas, imprimir una foto en

concreto acercando una cámara a una impresora o un documento PDF en un teléfono móvil con NFC, póster o anuncios con NFC.

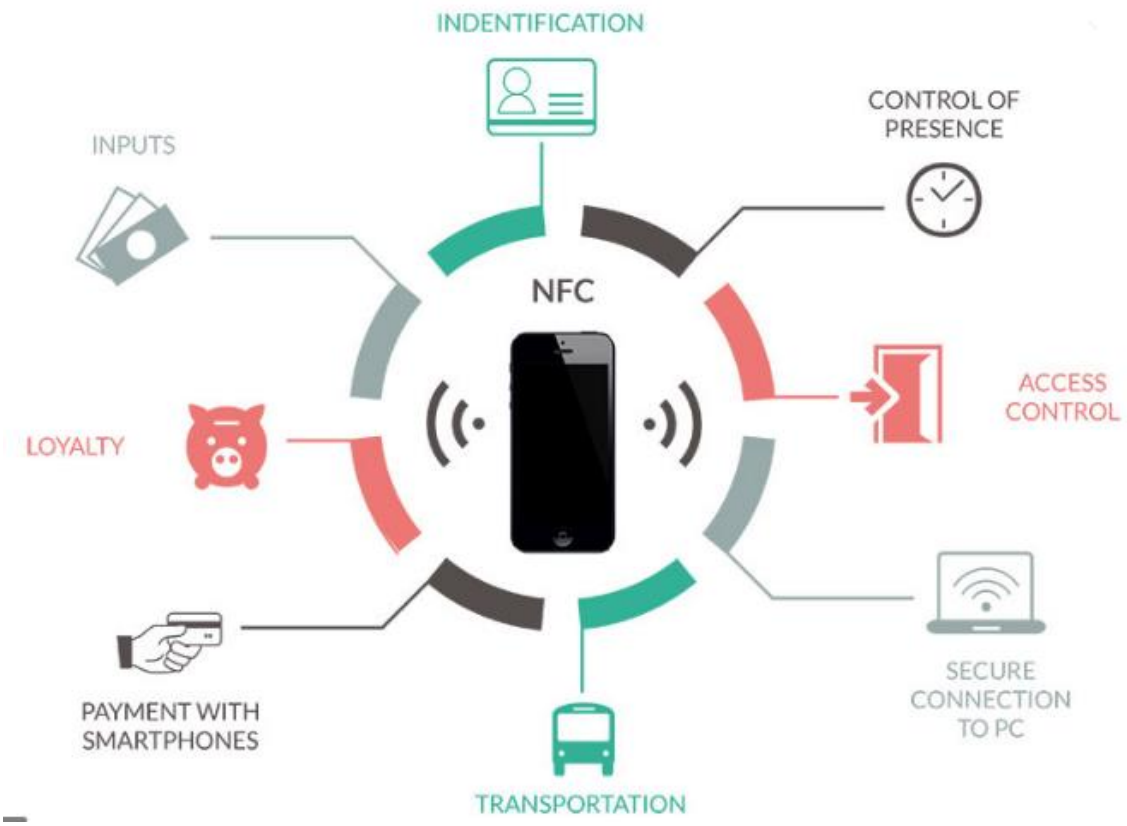


Figura 27 Aplicaciones de Dispositivo NFC

Criptografía en NFC

Comunicación, autentificación y seguridad en NFC

Una de las razones por que el protocolo NFC, es más seguro que otros protocolos de identificación por radio frecuencia, es porque permite autentificación.

Mientras que identificación, nos puede dar todos los datos del tag que esta leyendo, no se nos garantiza que el tag, ni el usuario sean genuinos.

La autentificación necesita un paso extra en la seguridad de la tarjeta, y es por eso que este sistema de autentificación frente a la identificación se usa en sistemas de tickets, autentificación de documentos u objetos, control de acceso y más. El concepto es que este tag no puede ser copiadas.

En los chips menos seguros, el sistema de autentificación estaba embebido en el chip, y accedías a este con comandos de NFC. En este sistema había una firma "estática" directamente unida al identificador universal del chip (UID). Y aunque era difícil de acceder y de clonar, en los estándares actuales no se considera una "buena solución"

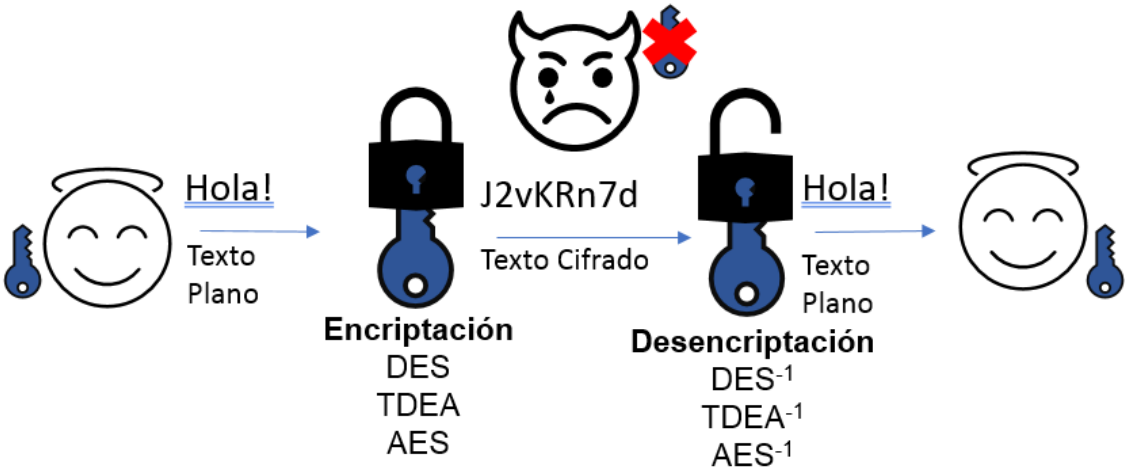


Figura 28 Criptografía simétrica

Los chips de nueva generación cuentan con dos avances. Generan un código único cada vez que se comunican (OTP, one time password), o utilizan un protocolo de desafío-respuesta

La criptografía es la ciencia que estudia los protocolos, algoritmos y distintos sistemas para dotar de seguridad a la comunicación y a los datos, y que ninguna parte ajena a esa comunicación pueda entrometerse. Estos sistemas reciben el nombre de criptosistemas.

La criptografía comenzó a tomar importancia durante la segunda guerra mundial, donde el bando alemán usaba la máquina ENIGMA para encriptar y desencriptar mensajes y así evitar que fueran capturados por el bando aliado.

De los errores del pasado, y de cómo se llegó mediante ingeniería computacional a resolver la criptografía de ENIGMA, podemos definir la mejor forma de cifrar un mensaje como aquella en la que mensaje cifrado y mensaje original tengan el mínimo posible de información relacionada, y el nivel de entropía, que refleja la incertidumbre de la generación de la clave, y que aun conociendo parte de este resultado sea imposible deducir el restante.

Protocolo desafío-respuesta o Handshake

Reciben este nombre los protocolos que permiten autentificación siguiendo el siguiente sistema.

El verificador, presenta un desafío, y se lo envía al verificador.

La parte que se autentifica, tras recibirla elabora una respuesta.

El verificador comprueba que la respuesta es correcta, y la parte pasa a estar autentificada.

Normalmente estos protocolos son de autentificación mutua, ambos tienen que responder.

Un ejemplo sencillo y anticuado es el uso de una contraseña numérica. El verificador preguntaría, ¿Contraseña? Y la respuesta es un valor numérico. En este caso un atacante que escuche podría autenticarse respondiendo esa contraseña.

Si establecemos un sistema de llaves públicas, un atacante no podría escuchar la clave secreta que ambos conocen

Usando algoritmos criptográficos, que la clave publica permanezca en secreto es muy importante. Para ello se busca que esta llave se use lo mínimo posible en entornos en los que pueda verse atacada. Aunque un mensaje este cifrado si un atacante, conoce el mensaje que se envía, podría descifrar esta key. Por eso se busca que siempre que haya una comunicación, sea entre dos dispositivos que ya tienen acceso a esta key pública.

Para mostrar que ambos tienen acceso a esta key publica, el desafío que se envían consiste en crear un numero pseudo-aleatorio (Mifare usa LFSR) para generarlo. Este número pseudo aleatorio(nounce) se encripta con la llave pública. El challenge que ha de resolver es la desencriptación de este número pseudo-aleatorio. En nuestro caso, si el número que envía es correcto, el otro dispositivo realiza el mismo desafío. Tras realizar esta autentificación mutua, la comunicación comienza, siguiendo el protocolo de encriptación.

Explicación de los algoritmos criptográficos

Existen dos tipos de algoritmos criptográficos, simétricos y asimétricos. Para que estos algoritmos funciones es necesario ponerse de acuerdo en la forma de comunicarse.

Los algoritmos simétricos reciben este nombre porque utilizan la misma clave, para encriptar y desencriptar el mensaje. Tanto el que envía los datos, y los encripta, como el dispositivo receptor, comparten la misma llave. Para que esta comunicación sea segura, la clave tiene que mantenerse en secreto, ya que, si se descubre o se filtra, un atacante puede ganar acceso y desencriptar el mensaje. En NFC se utiliza el Estándar de Encriptación Avanzado (AES).

Los algoritmos asimétricos utilizan distintas claves para encriptar y desencriptar el mensaje. En este caso la llave de encriptación puede ser pública y la lleve para desencriptar privada. En este caso, ambos dispositivos comparten su clave pública. Para comunicarse, el dispositivo que encripta el mensaje lo hace usando la llave pública del dispositivo al que lo envía. Cuando la recibe, este dispositivo desencripta el mensaje con su clave privada.

Un atacante podría, conseguir acceso a una clave pública. Esto no afectaría a la integridad ni a la seguridad del mensaje, ya que esta clave solamente sirve para encriptar el mensaje, no para descifrarlo. En cambio, con una llave privada, los mensajes sí que estarían comprometidos ya que se podrían desencriptar por el atacante. En NFC encontramos el algoritmo asimétrico Rivest, Shamir y Adleman (RSA) y el algoritmo de curva elíptica.

Estos algoritmos, tanto como RSA como AES se pueden implementar tanto por diseño de hardware, como por software.

(102190)

AES, Protocolo Criptográfico Simétrico

Dentro de los protocolos de encriptación simétrica encontramos varios ejemplos, entre ellos Serpent, DES, Safer, Des. De estos los que más prevalencia han tenido en el ámbito del NFC han sido TDES y AES-128, siendo este último el usado por Mifare.

El AES es un sistema de encriptación simétrico, en el que las keys o claves pueden ser de 128,192 o 256 bits. En NFC se utilizan llaves de 128 bits. Teóricamente a más bits más segura es frente a ataques de fuerza bruta (probar combinaciones de manera aleatoria).

Este algoritmo se introdujo en 2001 y todas las amenazas contra este de momento han sido teóricas, lo que significa que el tiempo para resolver este sistema, está muy por encima de lo que los ordenadores actuales pueden manejar.

El principio básico de AES es que cada unidad de datos, que recibe el nombre de texto plano o plaintext, se remplaza por un dato distinta. La razón por la que AES es tan seguro es por que utiliza un proceso de expansión, en el cual la key inicial da lugar a nuevas sub-keys llamadas roundkey, estas han sido generadas a partir de sí misma a través de varias rondas de modificación, lo que hace más difícil romper la encriptación. El resultado de esto da lugar una forma de encriptación muy sofisticada, que puede implementarse tanto por hardware como por software.

AES 128, y sus variantes son virtualmente imposibles de descubrir usando métodos de fuerza bruta. Con la tecnología actual de computación se tardarían billones de años. Al ser tan seguro, los únicos ataques posibles son en sistemas con AES mal configurado.

La encriptación por este método tiene 6 pasos, y la desencriptación es este método, pero siguiendo los pasos al revés.

El primer paso es la expansión. 128 bits son 16 bytes, que se colocan en una matriz de 4x4. Cada columna o fila esta formadas por bloques de 4 bytes, llamados words. Esta matriz recibe el nombre de state.

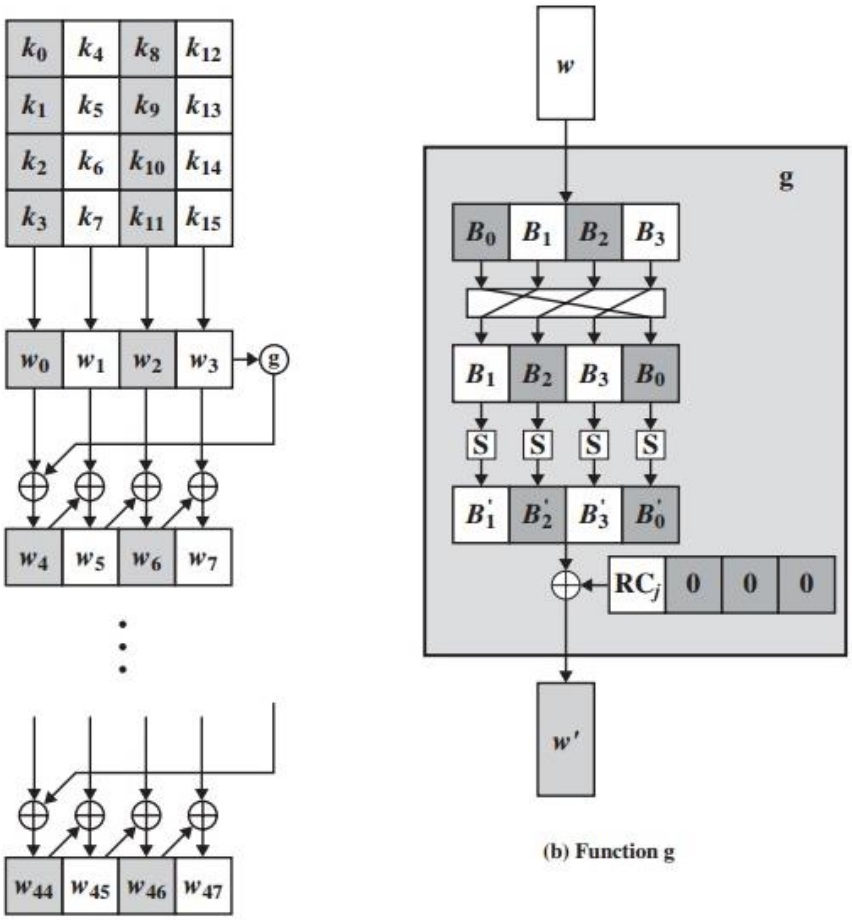


Figura 29 Generación de las 48 Words del algoritmo simétrico

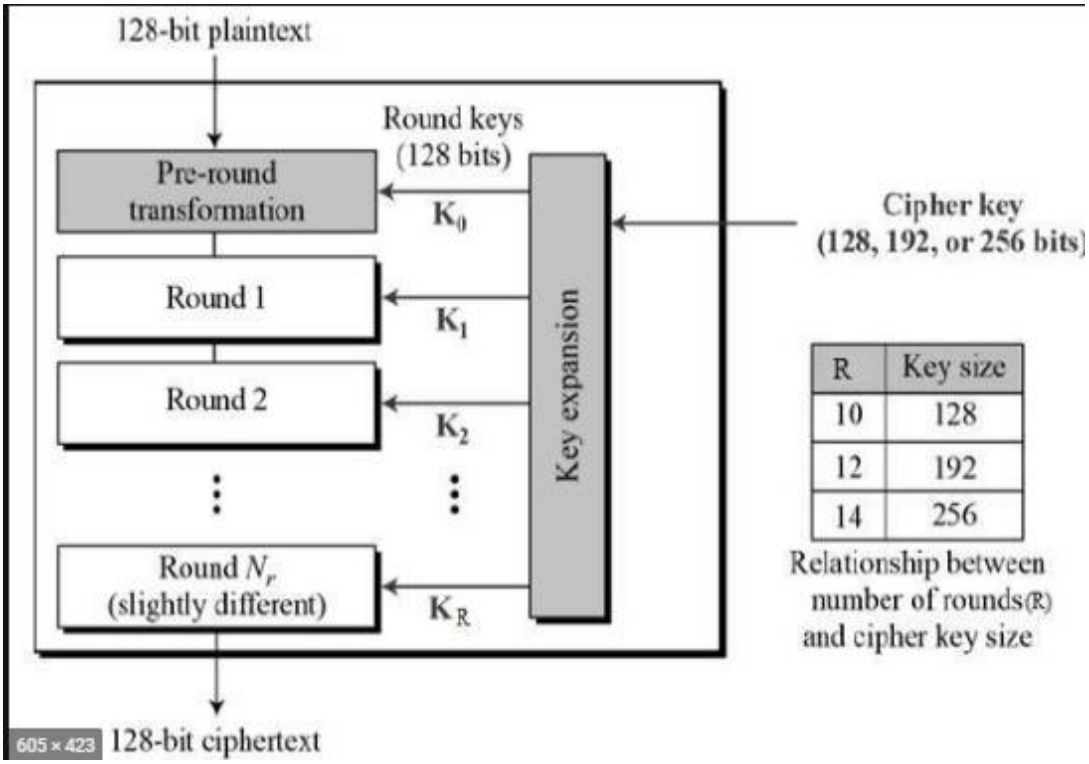


Figura 30 Expansion de clave a partir de N rondas.

Cada ronda, procesa una matriz de entrada (input) y genera una matriz output. Se produce una iteración 10 veces, por eso se conoce como cifrado de bloques iterado, donde el plain text, texto plano.

El proceso AES requiere 6 pasos, que son los siguientes.

1) Expansión de la key, manipula la key y genera una round key. La clave publica se expande, tomando como input 16 bytes y generando una matriz lineal de 44 palabras (176 bytes)

2) AddRoundKey, se produce una suma entre la roundkey [1], una matriz formada por las palabras desde W0 hasta W4, y la matriz state, nuestro plain text.

   Ahora se realiza este proceso 10 veces o rondas más.

3) Sustitución. Usando la tabla de Rijndael, cada byte se sustituye por el byte que ocupa su lugar en la tabla.

4) DesplazarFilas = Un paso de transposición donde las filas se desplazan de manera cíclica, la primera no se desplaza, la segunda se desplaza de una en una casilla, la tercera de dos en dos y la cuarta fila de tres en tres.

5) Mezclar columnas, multiplica el resultado de desplazar filas con una matriz multiplicante. [Hacer esa foto con una formula]

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 02\ 03\ 01\ 01 \\ 01\ 02\ 03\ 01 \\ 01\ 01\ 02\ 03 \\ 03\ 01\ 01\ 02 \end{bmatrix} \times \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Figura 31 Matriz multiplicante para MezclarColumnas

6) Se repite desde el paso de AddRoundKey, 16 veces más.

Como funciona la expansion de key.

Key: 0f1571c947d9e8590cb7add6af7f6798 // Privada, ambos dispositivos disponen de ella para comunicarse.

Primero la expansion, usando la key, generamos 44 Words (w)

W0 = 0f 15 71 c9	Rotar(w3) = 7f 67 98 af= x1
W1 =47 d9 e8 59	Sub(x1) = d2 85 46 79 = y1
W2 = 0c b7 ad d6	Rcon = 01 00 00 00
W3 = af 7f 67 98	Y1 + Rcon1 = d3 85 46 79 = z1
	a
W4 = W0 + Z1	Rot W7 = x2
W5 = W4+ W1	Sub(x2) = y2
W6 = W5 + W2	Rcon = 02 00 00 00
W7 = W6 + W3	Y2+ Rcon2 = z2

Siguiendo esta formula, se realiza la expansión de la clave hasta generar W43.

La operación sub toma el input de 8 bit en 8bit, y cada uno lo sustituye por el valor que le correspondería en la Rijndael S-box, una tabla de 16x16. En esta tabla, si el valor introducido es por ejemplo 31(hex), el valor que sale es c7(hex).

AES S-Box

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Figura 32 Tabla Lookup del algoritmo AES

Las palabas W0-W3, formarían la primera Round Key, W4-W7 la segunda, hasta formar la round key numero 10.

RSA y ECC, Protocolo Criptográfico Asimétrico.

RSA es un algoritmo de encriptación que utiliza 2 tipos de key, publica y privada. En ocasiones a estos algoritmos también se los conoce como algoritmos de key pública.

Cuando un mensaje ha sido encriptado con una llave pública, este tan solo puede ser desencriptado con una key privada. La diferencia frente a los protocolos de clave simétrica es que tanto la encriptación como la desencriptación se hacen usando la misma clave privada. Esta diferencia convierte al RSA en el método de encriptación más seguro para comunicarse en situaciones en las que no existe la posibilidad de distribuir de manera segura las keys a los dispositivos.

Esta encriptación se usa en ocasiones con otros métodos de encriptación, y con las llamadas firmas digitales, que añaden integridad y prueban la autenticidad del mensaje. No se suele utilizar para encriptar mensajes, ya que es menos eficiente que los protocolos de encriptación simétricos.

Es por esto que, en ocasiones, un documento se encripta con un algoritmo de llave simétrica como el AES, pero es la llave simétrica la que se encripta con RSA. Es por eso por lo que tras esta comunicación tan solo la entidad que tiene

acceso a la clave privada RSA puede desencriptar la llave simétrica, que desencripta el archivo.

El método para encriptar el mensaje usando RSA recibe el nombre de intercambio de key Diffie-Hellman. Las bases de este algoritmo dictan que la publicación de la clave publica nunca compromete a la clave privada, de manera que no se puede descifrar esta, solo si se conocen tanto la clave publica como la privada. Además de durante la criptografía, la generación de estas contraseñas ha de tener la mayor entropía posible.

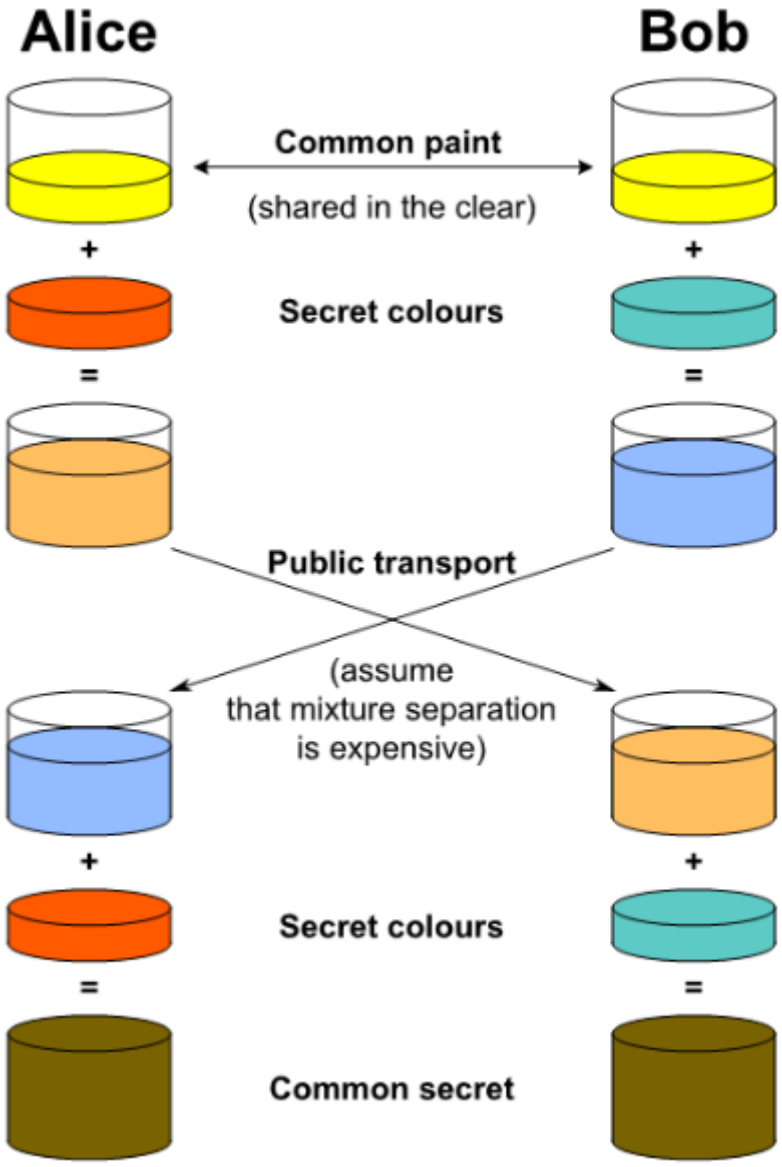


Figura 33 Ejemplo Visual del intercambio Diffie-Hellman

En esta explicación del intercambio Diffie-Hellman, Alice y Bob quieren comunicarse usando una clave secreta siguiendo protocolos simétricos, pero no tienen manera de asegurarse que no haya nadie escuchando esa clave, que aún no han intercambiado.

Primero establecen un color de base, el amarillo y un color que no se comunicaran el uno al otro, Alice genera naranja y Bob verde. En todo momento asumimos que no se puede descubrir que color se ha usado en la mezcla, ya la única manera es con fuerza bruta, pero suponemos que por la complejidad del color (solo sirve un naranja muy específico, #ff8128) no se puede obtener el color secreto a partir de la pintura que ha mezclado Alice ni Bob. Ahora si Bob y Alice mezclan el color que se han repartido, con su propio color secreto, obtendrán ambos el mismo color, que podrán usar como clave en un protocolo simétrico.

(Milanov, 2009)

El funcionamiento del RSA es el siguiente. Para la simplificación, usamos un mensaje (texto plano) M como un entero. Un texto cifrado C. E es la clave pública y D la clave privada, también números enteros. (E y D comparten un módulo n)

Para encriptar M, elevamos ese número a la potencia E.

Para desencriptar C, elevamos ese número a la potencia D.

$$C = M^E \ de\ modulo\ n.$$

$$M = C^D \ de\ modulo\ n$$

El tamaño de la información se mantiene debido a que los módulos son similares.

Para que esto se cumpla, hemos de generar dos claves que cumplan esa función y que los números que las formen sean los suficientemente grandes como para que los medios de computación actual no puedan sacar D, a partir de E. El método matemático que explica cómo funciona esta generación de contraseñas de puede encontrar en los anexos. [ANEXO Yevgeny]

Otro protocolo de clave no simétrica es el ECC, o Criptografía de Curva Elíptica. Sigue el mismo caso que el intercambio Diffie-Helman, en el que los que se comunican combinan sus mensajes con un secreto seguro que es teóricamente imposible para un atacante de decodificar matemáticamente, sin conocer el secreto de uno de los dos. (Schneider, 1996)

Se define como curva elíptica el conjunto de puntos de números del plano real que cumplen la ecuación

$$y^2 = x^3 + ax + b$$

Con todos los puntos que se encuentran en la curva plana que genera y la operación de suma, que se define a partir de 2 puntos que forman parte de esa solución. Estos puntos son P(x, y) y Q(x,y). Sin entrar mucho en la matemática, en el algoritmo RSA se basaba en la dificulta de factorizar números primos, la dificultad de este reside en lo que llama problema de logaritmo discreto de la curva elíptica. Que básicamente es el problema que tiene en moverse desde un punto de esa curva a otro, ya que aunque conozcas la ecuación de la curva y el

punto en el comienzas, no conoces cuantos puntos recorres por medio. Esta función se conoce como suma de puntos.

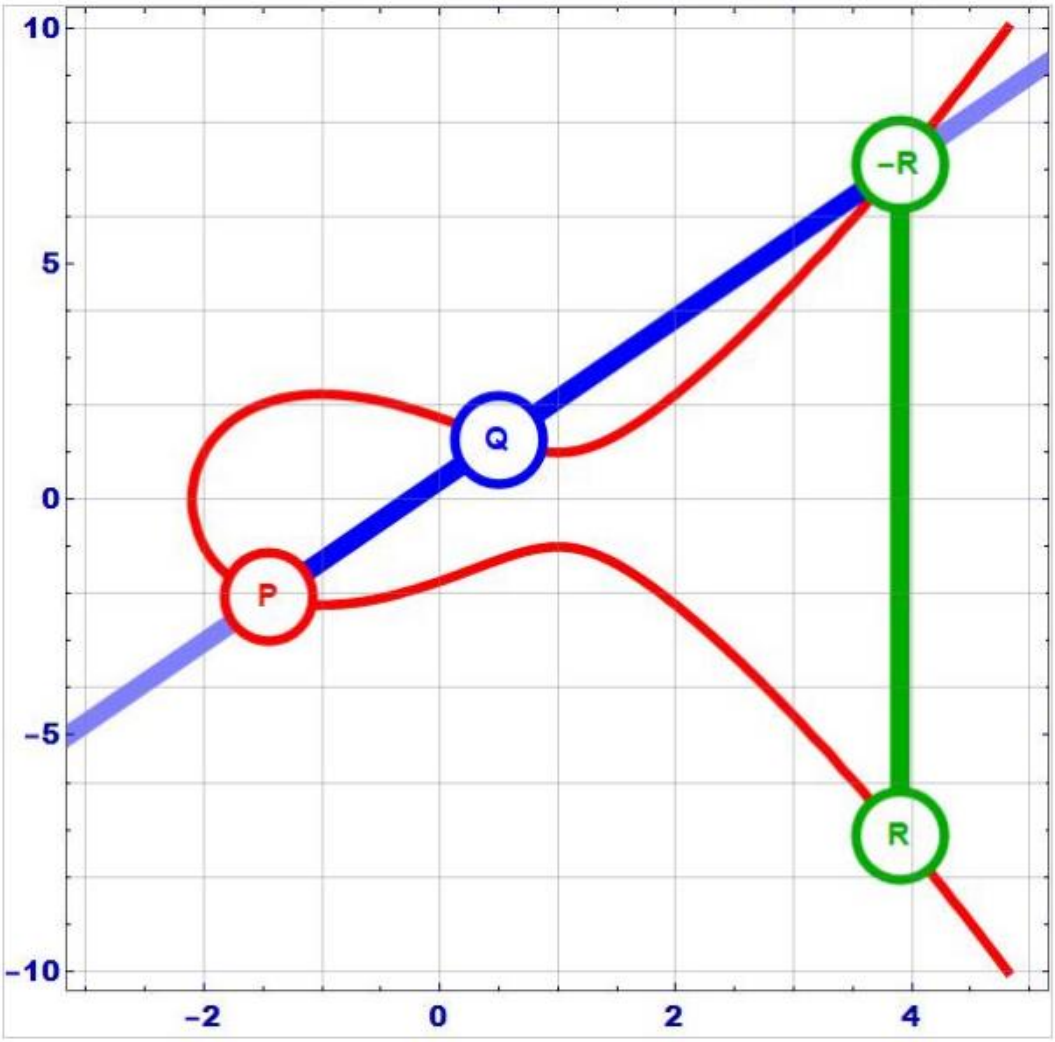


Figura 34 Ejemplo visual de criptografia ECC

P y Q son dos puntos de la curva que si se “suman” dan lugar a R. Si el punto Q, lo definimos como “estático”, y P forma parte del conjunto de la suma con Q, ha de existir un numero infinito (tanto como nos permita el campo finito definido) de números R que cumplan P+Q= -R. Y aun no existe ningún algoritmo eficiente para calcular el valor de R

Para implementar esta curva en el sistema Diffie-Hellmann, ambos dispositivos usan la misma curva y establecen su propia clave privada, además un punto común P1, Y a partir de este mismo punto común, número secreto, que será su clave, ahora cada uno multiplican el punto P1 por sí mismo un numero K de veces. Esta K es la clave privada de cada uno de ellos, K1 para el primer dispositivo y K2 para el segundo. Tras realizar la multiplicación K1 veces les quedara un numero -R1 y -R2.

Se cumplirá que R1=K1*G y R2=K2*G. Estos dispositivos se intercambian R1, y como hemos explicado antes, no existe ningún algoritmo eficiente para calcular K1 conociendo R1 y G, en caso de que un atacante oyera R1 y R2.

Ahora ambos multiplicarán este número recibido el número de veces que sea su clave K. y ambos conseguirán el mismo número secreto S, ya que S=K1(K2*G) = K2(K1*G). Este S es el secreto común que un atacante no puede descifrar, y lo pueden usar como clave para realizar una encriptación simétrica.

$$\exp\sqrt{\ln n \cdot \ln\ln n} = n^{\sqrt{\ln\ln n \div \ln n}} = (\ln n)^{\sqrt{\ln n \div \ln\ln n}}$$

Figura 35 Formula de relación de n, donde N es un número primo

SCP, Protocolo de Establecimiento de Canal Seguro.

El SCP, o protocolo de canal seguro, es una familia de protocolos que emplean varios mecanismos, sistemas y algoritmos para poder lograr una comunicación bidireccional, que busca evitar fugas de datos que podrían ayudar a resolver el criptoanálisis de un posible atacante.

En este caso estudiaremos el protocolo de canal seguro 03, ya que es el usado en las tarjetas MIFARE

SCP 03, Secure Channel Protocol

El SCP 03 es un documento que propone un nuevo protocolo basado en claves AES. Método que utiliza se denomina Encrypt-Then-Mac

El Secure Channel 03 o canal seguro se utiliza para “personalizar” las tarjetas NFC, durante la emisión y después de la emisión. Este protocolo permite generar “scripts” offline y datos para programar las tarjetas, y programarla sin ninguna conexión online con la entidad que genero esos scripts.

Si el personalizar esta tarjeta, por seguridad incluye alguna clave criptográfica, la clave “de transporte” ha de ser tan segura como la clave que se transmite. De esta forma, por ejemplo, una clave AES-192 podría usarse para transportar otra clave AES-192, pero una AES-128 no puede transportar, AES-192 o RSA-15360.

Este protocolo, si se cumple se genera una autentificación mutua entre ambos dispositivos. La manera en la que se consigue esto, es que desde manufacturación/emisión las tarjetas cuentan con varias claves de encriptación.

Estas llaves son obligatorias en las tarjetas, que no se modifican y que mantienen su seguridad debido a que la entidad que las ha creado, lo ha hecho usando un TRNG, o un generador de números aleatorio real. Estas claves son:

ENC: Una clave obligatoria que genera claves de sesión (S-ENC) para encriptar o desencriptar datos.

MAC: Genera claves de sesión (S-MAC), para la autentificación de canal seguro.

Las claves de sesión se generan siempre que se inicializa un canal seguro, y se utilizan en proceso de mutua autentificación. Estas claves de sesión son dinámicas y se borran tras finalizar el proceso de autentificación.

Tanto la tarjeta como el dispositivo que busca inicializar un canal seguro con la tarjeta, han de resolver el desafío criptográfico que se emiten simultáneamente.

El desafío es el siguiente;

8 bytes de desafío se calculan usando una derivación de datos de la clave de sesión, con el uso de números pseudoaleatorios. Si el dispositivo que lee la tarjeta resuelve este desafío (porque conoce la clave ENC, y la tarjeta resuelve también el desafío emitido por el dispositivo, entonces se produce una autentificación.

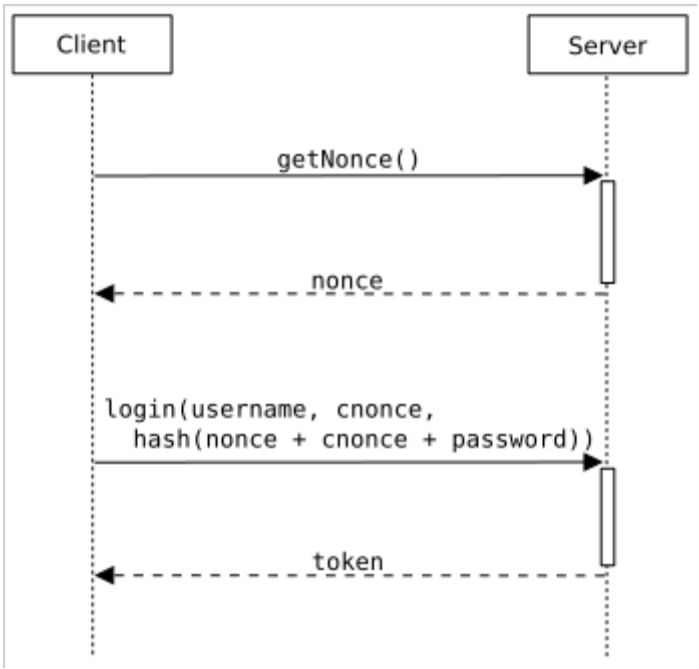


Figura 36 Protocolo SCP para generar token web

Ahora, todas las comunicaciones se cifrarán usando la S-MAC, que actuara con criptografía privada (AES, pero otros métodos de criptografía están soportados por distintos tipos de tarjetas).

Las normas del protocolo son las siguientes.

El nivel de seguridad del canal se ha de mantener durante la sesión a no ser que se modifique por la aplicación.

Si el nivel de seguridad se reduce a 0, o lo que es lo mismo, canal no seguro, el comando que este siendo transmitido se rechazara por el otro dispositivo que forme parte de la comunicación mediante la transmisión de un mensaje error.

El nivel de seguridad de los comandos se han de comprobar independientemente de si se ha establecido un canal seguro o del nivel de seguridad.

Si el nivel de seguridad del mensaje es mayor que el nivel de seguridad establecido, el canal seguro se cortará, y se establecerá un nivel de seguridad 0, y se transmitirá un mensaje de error. En cualquier otro caso, la aplicación es responsable de procesar ese comando.

La sesión se terminará, y el nivel de seguridad se reducirá a 0 siempre que :

Se intente inicializar un nuevo canal seguro.

Se termine la Sesión por parte de la aplicación.

Se produzca un Power Off en el tag. Especificado por la aplicación **(GlobalPlatform, 2009)**

SAM, Modulo de Acceso Seguro y SE, Elemento Seguro

Un Elemento Seguro, (SE), es una combinación de hardware, software, interfaces y los protocolos que, embebidos en un sistema o en un dispositivo proporcionan almacenamiento seguro.

La definición que global platform da a elemento seguro como, una plataforma o hardware resistentes a manipulaciones y diseñada para almacenar datos confidenciales y datos criptográficos como claves, siguiendo los requerimientos de seguridad. Cuando una aplicación NFC requiere un alto nivel de seguridad, como por ejemplo sistemas de pago, las claves para la comunicación se almacenan en estos chips denominados elementos seguros. En caso de una aplicación de pago, por ejemplo, también se almacenan todos los datos personales, ya sea número de cuenta, fecha de caducidad etc.

Un elemento seguro es un chip que se encuentra en los dispositivos con NFC.

La arquitectura típica de estos chips son una CPU, un OS, RAM, ROM y EEPROM, timers, “cryptoengine”y puertos de comunicación.

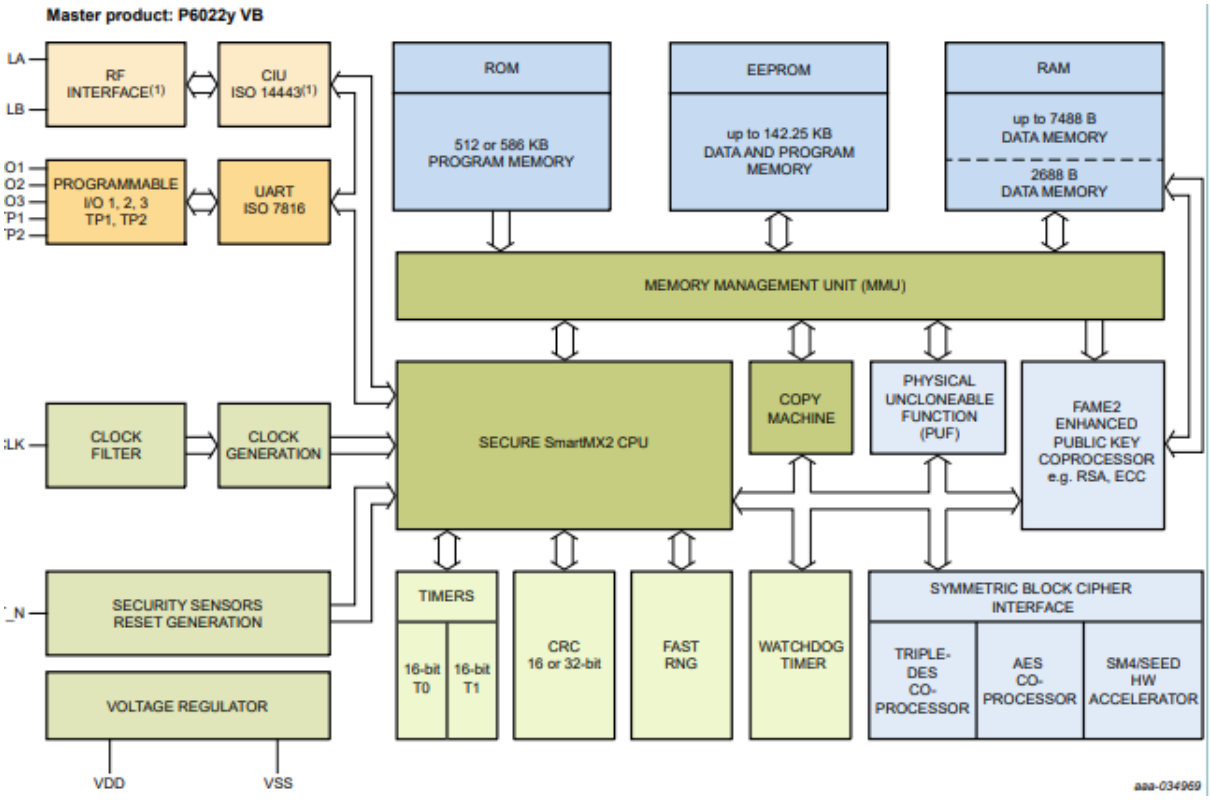


Figura 37 Hardware de un SAM de tarjeta JCOP

Las aplicaciones dentro del elemento seguro se encargan de varias tareas, como el protocolo desafío-respuesta, autentificación, filtrado de datos, derivación de claves, etc. Para ser más específicos el elemento seguro define un ambiente en el que se pueden ejecutar las aplicaciones de comunicación.

Existen varias formas y alternativas para la creación de un elemento seguro. Las más comunes en comunicaciones NFC son.

SE Embebido, que consiste en un chip microcontrolador que se encuentra embebido en la placa madre. Estos chips se organizan y se programan por el OEM, o la empresa manufacturadora.

Al encontrase embebido, el consumo de energía y la eficiencia son mayores.

Si se busca utilizar un dispositivo NFC como un teléfono móvil y queremos que emule una tarjeta NFC, los datos se almacenan en un Secure Element UICC, en el caso de un móvil, una tarjeta SIM.

UICC Secure element, o tarjeta de circuito integrado universal. Un ejemplo de las tarjetas UICC son las tarjetas SIM, (Modulo de identificación de Suscriptor) usadas en teléfonos móviles para tecnologías 3G. La principal diferencia es que tiene varias aplicaciones extras más allá del protocolo LTE y almacenar datos y PIN que tienen actualmente las SIMs. Un módulo de acceso seguro, o SAM, es como se denomina a la solución hardware para añadir a un sistema de key-management (KMS), o gestión de claves.

(Sonal Rohilla, Syscom Corporation, 2015)

En este caso el SE se encuentra embebido en el UICC, y es este circuito es el que se comunica con el controlador NFC de manera directa. Esta comunicación directa sigue varios protocolos soportados por la mayoría de los operadores, lo que reduce el precio de desarrollo de aplicaciones. Como el UICC se puede mover de un dispositivo a otro, el SE y las aplicaciones unidas a él permanecen, lo que le da portabilidad. En caso de que se pierda, si no se conoce la clave de acceso al UICC estos datos se destruyen.

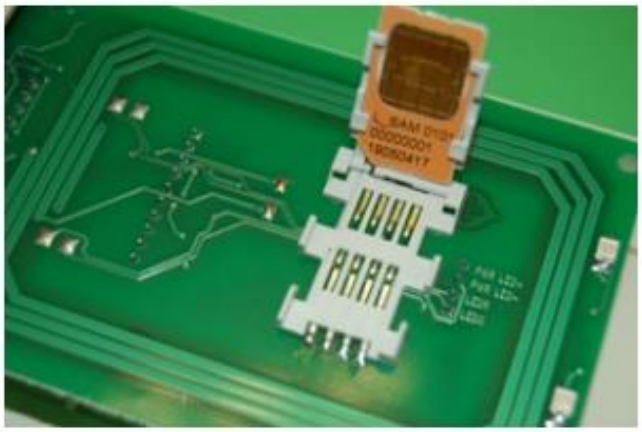


Figura 38 Adaptación de un UICC a una PCB de RFID

Un ejemplo de tarjetas que usan este tipo de tecnología son las JCOP, o Java Cards. Estas tarjetas tienen aplicaciones basadas en Java para funcionan el Elementos Seguros del tipo UICC. Esta tecnología es la que se utiliza en sistemas de identificación para el gobierno, mercado de pagos y transacciones, donde estas tarjetas se usan como tarjetas de crédito, en unidades que superan los cientos de millones.

El contenido de la Applet está completamente aislado mediante la arquitectura del sistema operativo, mediante Firewalls, MMU, contadores seguros, sensores anti-tampering, etc.

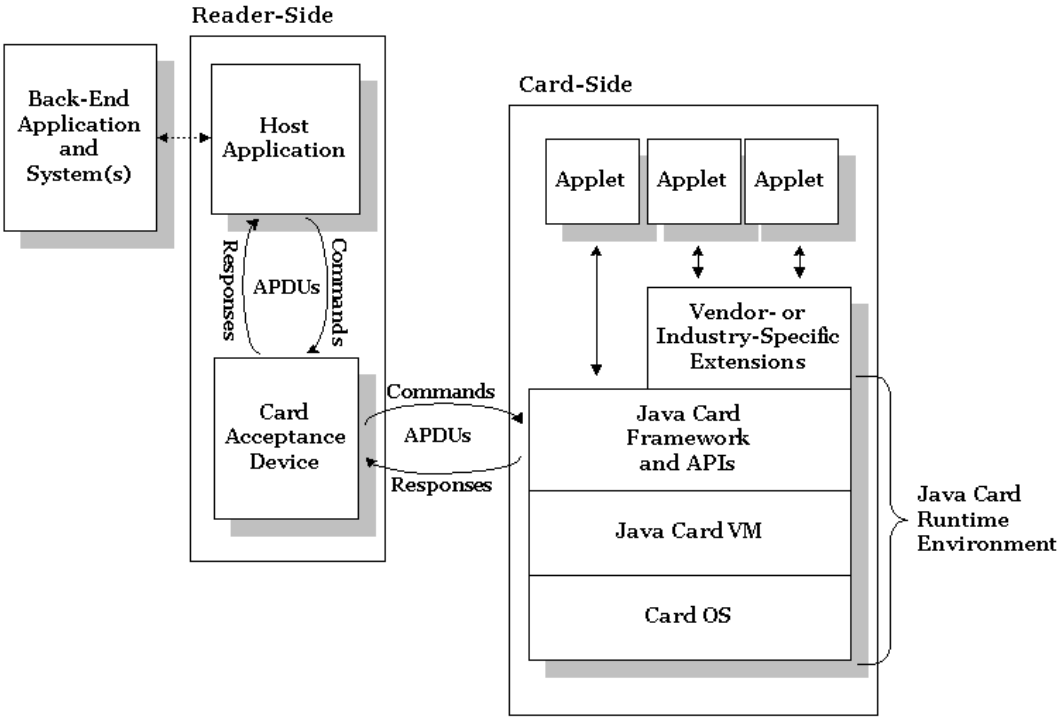


Figura 39 Diagrama de Bloques de comunicación y contenido de una Smart Card JCOP

La manera de comunicarse con el SE050, se realiza con el protocolo APDU, Aplication Protocol Data Unit, muy similar al NDEF, ya que los mensajes que se envían están compuestos de encabezado y contenido. Si se trata de una tarjeta Smart como las usadas en bancos, mediante los pines de contacto o contactless con el protocolo NFC. En el SE050, se transmiten las APDU mediante I2C. El formato de las APDU se encuentra en el ISO 7816, donde se específica header contiene flags y longitud, y el cuerpo o contenido contiene los datos.

El SE050 realiza las operaciones criptográficas, y almacena claves y archivos. El sistema de Applets de JavaCard utiliza sesiones para diferenciar entre usuarios. Este usuario puede usar, o un código o un ID dentro de una tarjeta NFC o una Smartcard para autentificarse en el sistema, dependiendo de la interfaz que se haya instalado. Además, esta preparado para realizar operaciones seguras con los productos de Mifare, y protege la clave privada utilizada para establecer el canal seguro con una tarjeta NFC de Mifare.

Es capaz de realizar todas las funciones de derivación, autentificación y generación de claves de sesión.

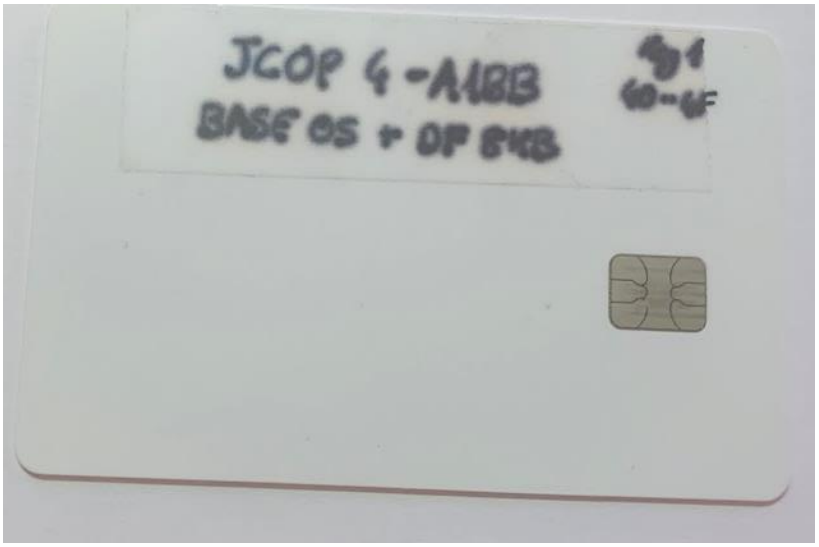


Figura 40 Tarjeta JCOP para desarrolladores

Es importante también tener en cuenta que las tarjetas JCOP y EMV, que son tarjetas generalmente usadas para transacciones de crédito, formadas a partir de un SE integrado en la tarjeta, si se sigue un protocolo erróneo esto puede dar lugar a un posible ataque Man-in-The-Middle, donde un chip se sitúa encima de la tarjeta, y modifica algunos datos o comandos que no estén cifrados. **(Houda Ferradi, 2010)**

ISO 14443-A - MIFARE

Existen una serie de estándares, que definen los protocolos de comunicación y formatos de intercambio de datos RFID y NFC.

En la actualidad el 80% de las aplicaciones del mercado están controladas por el modelo de tarjeta MIFARE Classic, que ha evolucionado en los modelos Mifare Ultralight, DesFire y Plus.

Figura 41 Tarjetas y Tags Mifare Classic Comerciales

Este chip es una memoria EEPROM con procesadores criptográficos. La memoria EEPROM puede ser de 1K o de 4K según el modelo. Los datos se guardan en bloques de 16 bytes. En total hay 16 sectores con 4 bloques cada sector en el caso de 1K.

Salvo el primer bloque, que contiene la ID de la tarjeta y los datos de manufacturación.

En el resto, cada sector usa un bloque de datos para almacenar las llaves A y B, junto con los bits de acceso, que son 4 bytes donde se encuentras los bits que indican las condiciones de acceso para cada una de las operaciones de los otros 3 bloques de datos restantes. Con estos bits, por ejemplo, si se encuentran todos a 0, para acceder a ese sector hace falta la llave A o la B, para cualquier acción ya sea leer, escribir o transferir datos. La tabla de verdad es la siguiente.

C1	C2	C3	read	write	incr	decr, transfer, restore
0	0	0	key A|B	key A|B	key A|B	key A|B
0	1	0	key A|B	never	never	never
1	0	0	key A|B	key B	never	never
1	1	0	key A|B	key B	key B	key A|B
0	0	1	key A|B	never	never	key A|B
0	1	1	key B	key B	never	never
1	0	1	key B	never	never	never
1	1	1	never	never	never	never

Figura 42 Tabla de posiciones lógicas para establecer si se puede escribir, leer o transferir ese bloque

En la datasheet, se explica cómo funciona. La tag comienza en el modo POR, que indica que está esperando un campo magnético que le de energía. El lector

entonces, puede enviar Request Estándar o Request All. En el request Estandar la tarjeta responde con un código dependiendo del tipo de tarjeta Mifare que sea.

Después se pasa al bloque anticolisión, donde si el lector detecta alguna, manda el comando AC. La tarjeta entonces envía encriptado su UID. Solo las tags cuya ID sea valida responderá, y el lector selecciona una tarjeta. La tarjeta respondera con un acknowledge.

Ahora el lector ha de especificar de que sector quiere leer, y utiliza la autentificación en 3 pasos. Esta autentificación es un protocolo desafio respuesta, en el que el lector usara una de las dos claves. El tag generara un nonce(Pseudo aleatorio) que codificara con la clave A, y el lector tambien lo hará. Tras recibir ambas respuestas, el lector enviara el comando leer y la tag devolverá los datos encriptados con la clave A.

Comandos en una comunicación NFC de Mifare Classic

Lector 26	REQA
Tarjeta 04 00	ATQA // Código de tarjeta Mifare classic1
Lector 93 20	Comando AC anticolisión
Tarjeta 9C 59 9B 32 6C	UID
Lector 93 70 9C 59 9B 30	SEL // Contiene el UID
Tarjeta 08 B6 DD	SAK (acknwoledge
Lector 60 01 F5 7B	Autentificar bloque 1 con la tarjeta A
Tarjeta 82 A4 16 6C	nonce de la tarjeta //Encriptado
Lector EF EA 1C DA	nonce del lector // Encriptado
Lector 8D 65 73 4B	respuesta
Tarjeta 9A 42 7B 20	Tag response
Lector 30 00 02 A8	READ // Encriptado
Tarjeta 9C 59 9B 32 6C 88 04 00 47	DATOS// Encriptado

Debilidades y Errores de Diseño

Uno de los errores de diseño de esta tarjeta es que este número pseudo aleatorio solo dependía del tiempo entre POR y el inicio de la comunicación.

Este número tiene una entropía muy baja, ya que el numero Pseudo-Aleatorio solo mide 16 bits. (65535) .

Un error que tenían también estas tarjetas NFC, es que si durante el protocolo de autentificación, cuando el lector envía su respuesta la tag checkea los bits de paridad de la respuesta antes de checkear si el contenido de la respuesta es correcto. Por eso si los bits de paridad son correctos(solo son 4), la tarjeta responderá con un código de error 0x05, error de paridad. Este código para que el lector lo lea, se envía encriptado. Como ya conocemos 0x05, podemos hallar 4 bits de la llave. Para facilitar el testeo y la integración de estas tarjetas al mercado, los manufacturadores mandan las tarjetas con claves por default, como por ejemplo AABBCCEEDDFF. En muchas ocasiones, estas claves no eran cambiadas por los encargados de las aplicaciones.

(Kasper, 2012)

Desarrollo de un Prototipo y Caso Practico

La tecnología NFC mueve millones de euros, esto la vuelve susceptible a ataques. Es por eso que muchas aplicaciones NFC tienen vulnerabilidades que las obligan a renovarse cada ciertos años.

Por eso, en el desarrollo de este proyecto, se ha buscado usar tecnología NFC que se considere state-of-the-art, especialmente en los aspectos que revuelven a la seguridad de la tecnología. NXP, compañía que desarrollo la tecnología MIFARE y fundo el Forum de NFC, ofrece una gran variedad de tarjetas, elementos seguros, lectores, chips, y sistemas preinstalados para desarrollar aplicaciones NFC.

Además también ofrece software y librerías para el desarrollo de aplicaciones y prototipos.

Placa Edgelock SE050, Como elemento seguro del prototipo.

El SE050 es una solución ready-to-use elemento seguro. Proporciona la raíz de seguridad a nivel de microcontrolador.

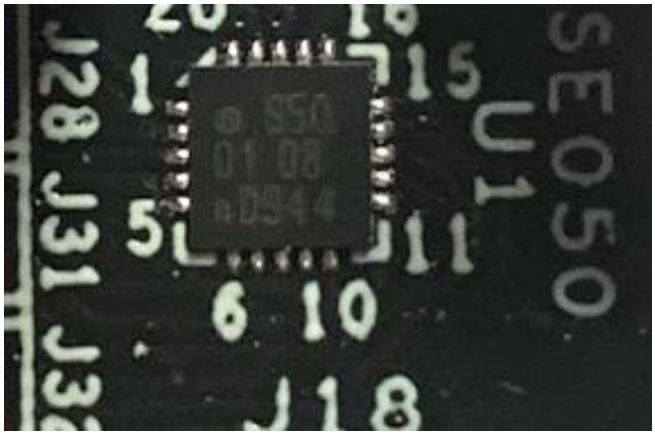


Figura 43 SE050 IC

Permite almacenar de manera segura credenciales y clave y realizar operaciones criptográficas en sistemas de comunicación y de control de acceso. Es versátil y además de poder usarse en autentificación, también permite usarse para interacciones con la nube y protección de datos de un sensor. Tiene seguridad de punto a punto.

Soporta algoritmos asimétricos como RSA y de Curva Elíptica. Tiene medidas de seguridad que protegen el sistema de todo tipo de ataques invasivos y no invasivos. Esta diseñado con un MiddleWare Plug&Trust de fácil uso.

Está diseñado para formar parte de un sistema IOT, unido a un controlador principal. Este controlador se comunicará con el SE050 mediante una interfaz I2C, donde el SE050 actuará como esclavo.

El SE050 también permite la conexión de una antena NFC, lo que lo convertiría en una interfaz contactless para tarjetas o teléfonos.

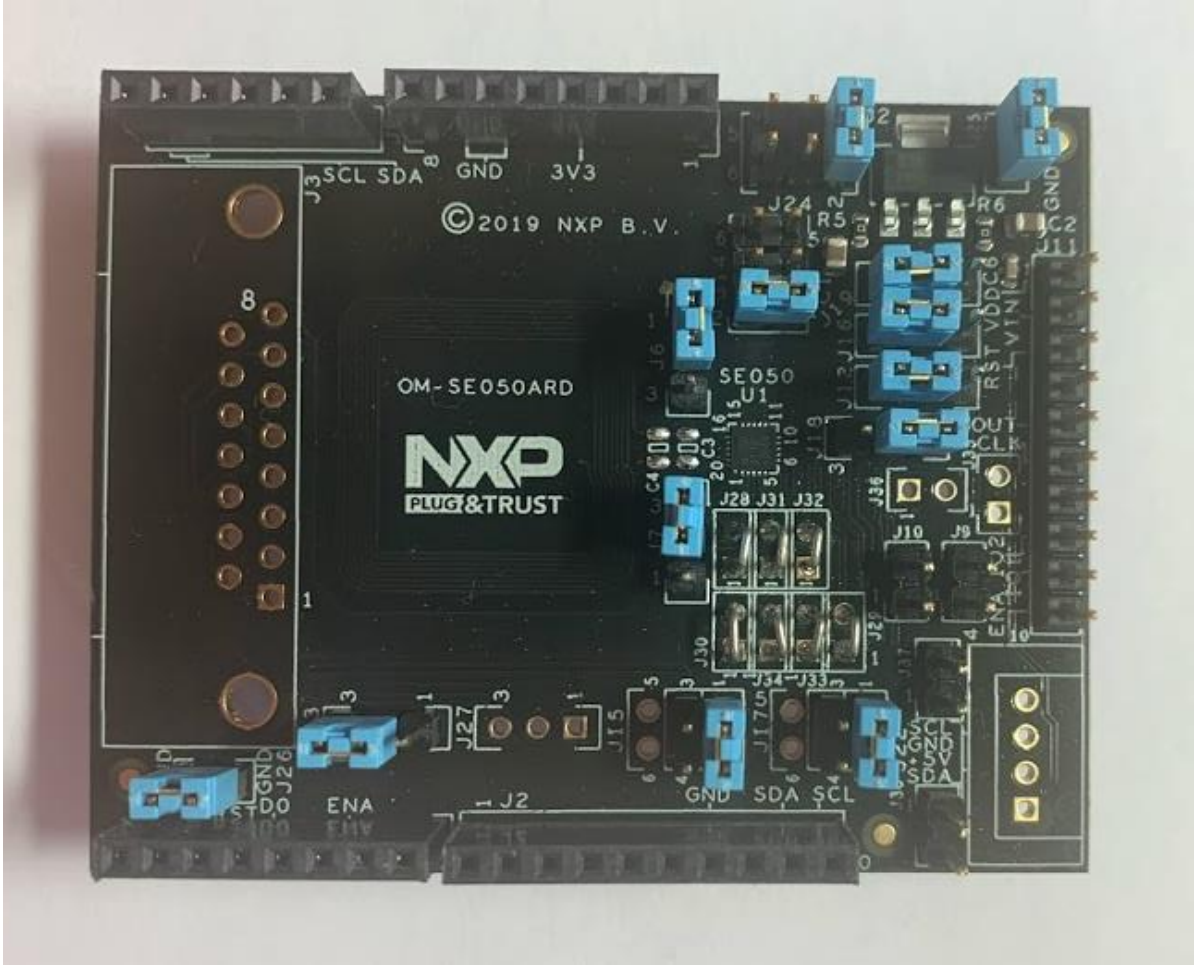


Figura 44 Tabla de desarrollo OM-SE050

El SE050 opera de forma completamente autónoma debido a que en su interior tiene un sistema operativo basado en JavaCard (JCOP4). Y es solamente este sistema el que puede acceder directamente a la memoria, lo que la deja completamente aislada del sistema de microcontrolador.

Además soporta SCP03 protocol, mediante encriptación de bus y de inyección de credenciales.

Los claves que almacena en el elemento seguro son las siguientes :

Symmetric Key (AES, 3DES), con operaciones de encriptación, derivación e importación segura.

ECC Key, con operaciones de firma(encriptación publica), de verificación, y generación de llaves de sesión.

RSA Key. con operaciones de firma, de verificación, y generación de llaves de sesión.

HMAC Key, con operaciones de inicialización de comunicación y finalización.

Binary File, valores que se quieran acceder con operación read/write autorizadas.

User ID, donde se permiten crear sesiones basadas en el ID de una tarjeta, en lugares donde haya varios usuarios y no sea necesario el uso de tarjetas de credencial (ID) con operaciones criptográficas.

Counter. Tanto para evitar tampering (reset tras un numero de intentos, como para el generador de números aleatorios). Las operaciones que permite son set, get y incrementar.

El SE050 complementa un sistema de control con otro microcontrolador, ya que inicia un Secure Channel con el controlador, y tiene un Software integrado que le permite usar los recursos de criptografía instalados en su Applet y sistema Operativo, y estos integradores se pueden acceder llamando a las APDUS del código desde el micrcontrolador principal. Por ejemplo “DiversifyKeyDF”.

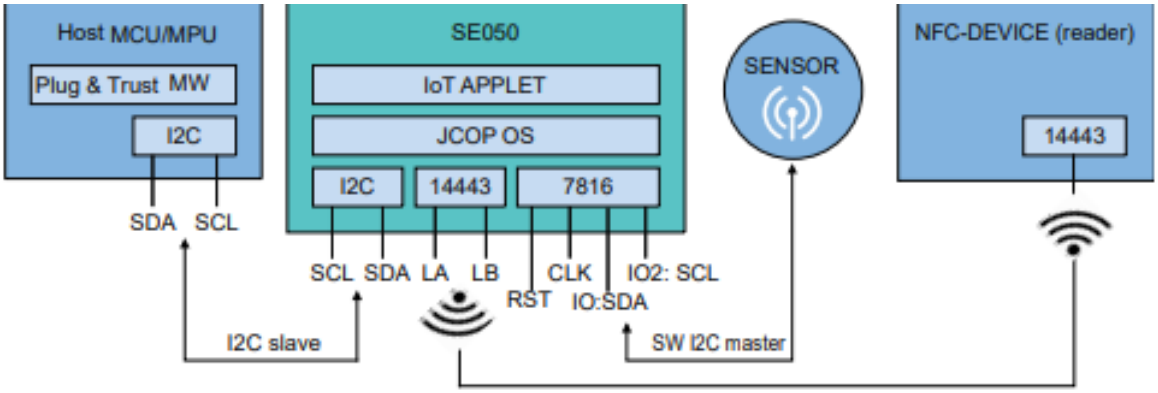


Figura 45 Esquematico de Conexion del SE050

El SE050 funciona como root-of-trust, o sistema base en el que confiar para la integración de claves. Está diseñado para lidiar con los estándares de seguridad en IoT e Industriales, al mismo nivel que la seguridad de tarjetas inteligentes actuales del mercado, usadas en banking y otras transacciones.

Debido a esto el SE050, se puede utilizar como inicio de un protocolo de canal seguro, para muchas aplicaciones como, por ejemplo, almacenar claves y credenciales para la autentificación de dispositivos y la conexión de estos a la

nube (Cloud). [https://www.nxp.com/docs/en/application-note/AN12401-SE050_secure_connection_google_cloud_iot_core.pdf]

CLRC663, como IC del lector.

El CLRC663 es un circuito integrado con multiprotocolo NFC diseñado para interactuar con usuarios.

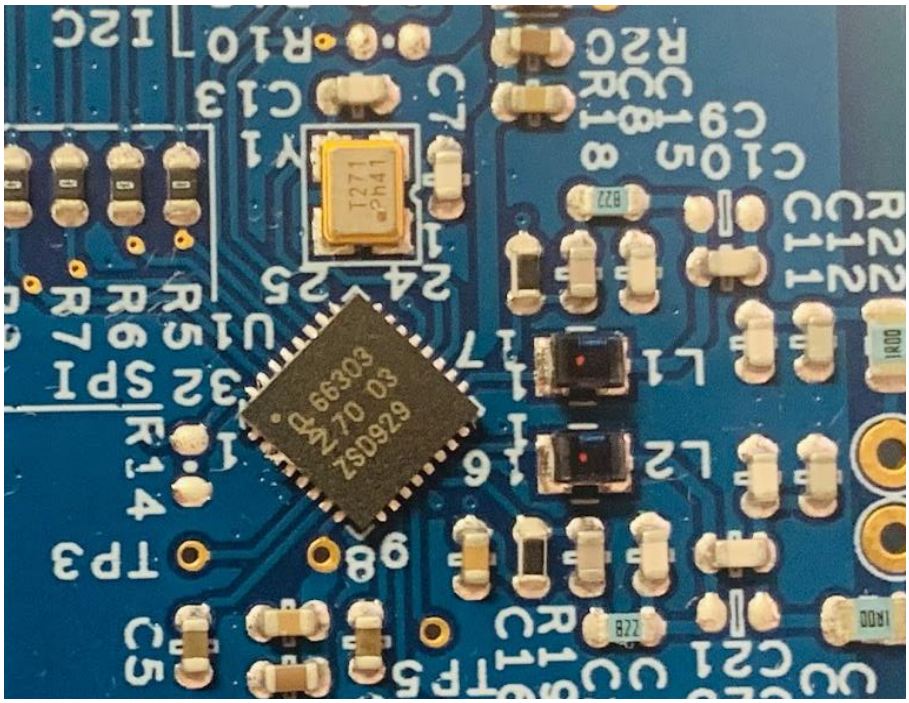


Figura 46 CLR663-03 IC montado en una placa de desarrollo

Multiprotocolo porque soporta :

NFC de ISO 14443 A, (Mifare Classic), Conexión Peer to Peer (ISO-18092), el Protocolo de FeliCa, entre otros.

Se puede usar en modo Lector, y como iniciador pasivo en el modo Peer-to-Peer.

Como comunicación con el Microcontrolador principal, soporta modos de comunicación en SPI, I2C y UART. También cuenta con una interfaz de I2C separa la implementación y conexión de un módulo de acceso seguro. (UICC o SAM),

Este SAM puede conectarse para operar como un co-procesador para las operaciones criptográficas. El microcontrolador CLR663, realizara una petición al SAM, generará la respuesta y la enviara a la interfaz I2C que llegara de nuevo al microcontrolador. Esta SAM, además también se encuentra conectada al

circuito integrado del lector, así que, en caso de necesitar, se proporciona una comunicación directa.

Mifare DesFire EV2, como tarjeta transponder.

La Mifare DESfire (MF3 IC D40), manufacturada por NXP, es una tarjeta RFID que cumple con la ISO 14443, operando a 13.56MHz.

Tiene un sistema de autentificación de desafío-respuesta, que se basa en AES-192. Tiene 4kB de memorita en la tarjeta que son capaces de almacenar 28 applets. Estas Applets pueden ser hasta 16 archivos cada uno de ellos asegurado con hasta 14 claves distintas. Cada tarjeta tiene una clave maestra o principal, y un protocolo que define cual es la autentificación necesaria para acceder a cada archivo.

Figura 47 Tres tarjetas Mifare DesFire EV2, 2 de desarrollo y una del Metro de Madrid

El protocolo de Mifare Desfire es el siguiente. El lector inicia la autentificación en protocolo de seguridad nivel 3.

En sistemas de Control de acceso, el sistema se basa en que existan 3 llaves, y cada una de ellas ha de inyectarse en el sistema siguiendo un protocolo de canal seguro. En el caso de Mifare, cada llave excepto la llave de identificación mutua,

se diversifican con un algoritmo similar al descrito en el protocolo de clave pública.

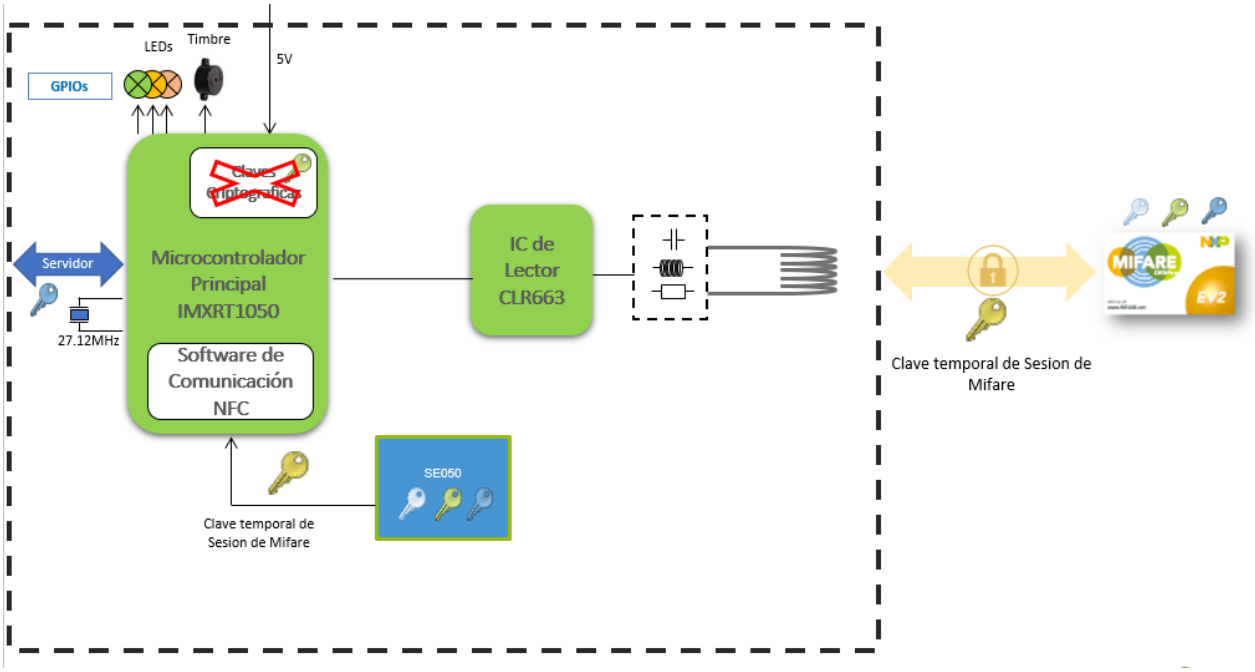


Figura 48 Esquemático de conexiones del proyecto.

APPMK o Master Key Application, esta clave esta almacenada en la tarjeta, y se utiliza para la personalización de los archivos y datos seguros.

APPVK, Validation Key Application, se utiliza para la validación y autentificación de los archivos que se comunican.

GMAK, General Mutual Authentication Key, esta clave se utiliza durante la autentificación mutua. Esta clave incluye un Identificador único que nunca varia, y que se utiliza como check de originalidad en algunos casos. Si no, esta clave se utiliza para la diversificación de claves, convirtiendo en las claves de cada tarjeta únicas. El algoritmo que se usa para diversificar puede ser AES, pero existen otros métodos explicado en este documento [AN10922]

La tarjeta que se ha utilizado en el caso practico es la Mifare DesFire EV2.

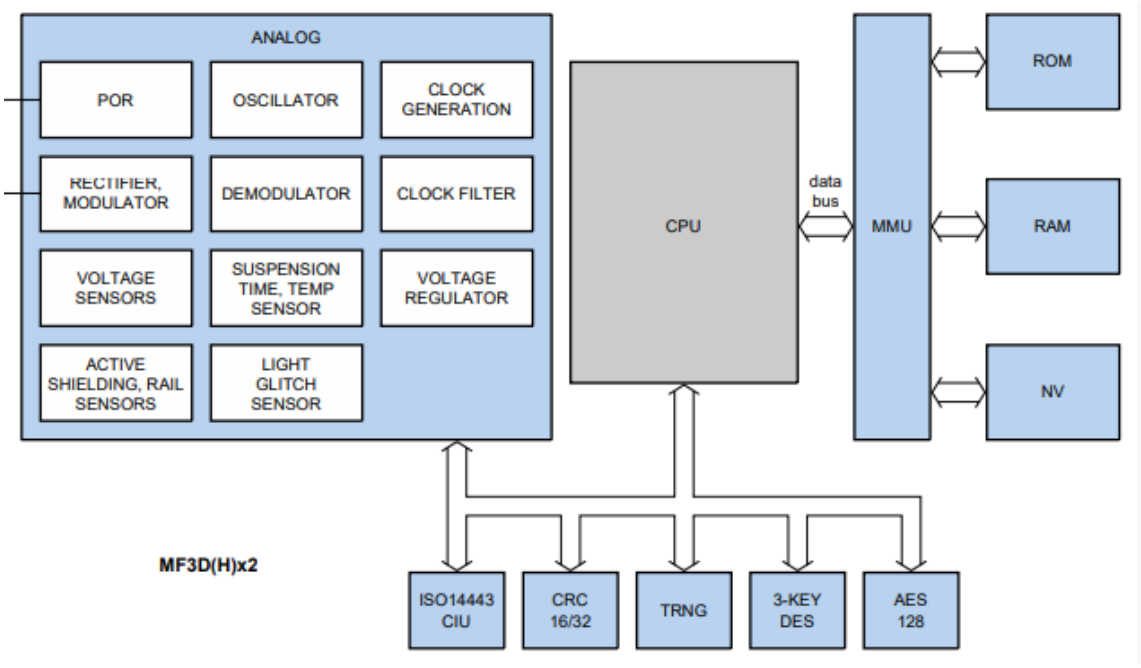


Figura 49 Esquema del IC de la tarjeta Mifare Desfire EV2

Frente a otras tarjetas Mifare como la Classic o la Ultralight, esta tarjeta permite ajustes para seleccionar el tipo de criptografía. Permite comunicación con D40 y EV1, para compatibilidad backwards, pero para nuevos proyectos, se recomienda usar el EV2.

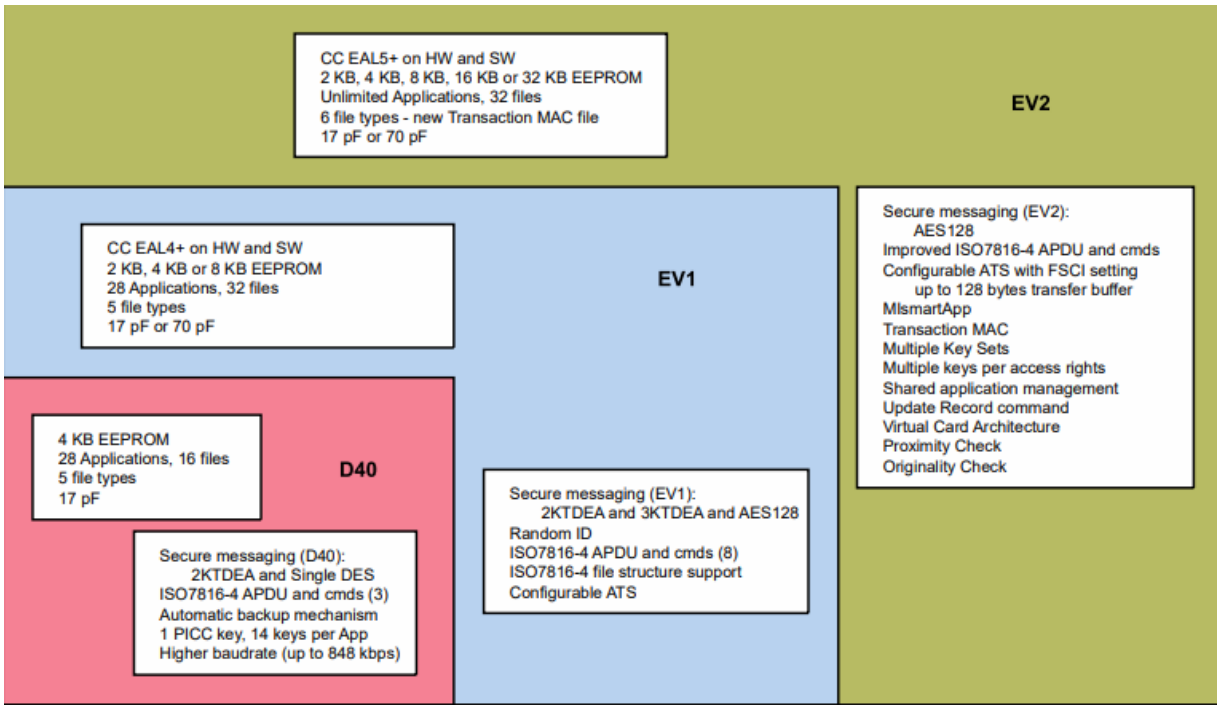


Figura 50 Distintas características de las 3 versiones de Mifare Desfire

También permite un sistema de Key Rolling. Con un simple comando, el lector puede solicitar a la tarjeta Mifare DesFire EV2 que modifique su set de claves, por otro de los que tiene almacenados. Puede almacenar hasta 3 sets de 14

claves. En caso de que las aplicaciones tengan permisos, se pueden intercambiar archivos entre dos aplicaciones.

SDM, Secure Dynamic Messaging. Si queremos usar la Mifare para almacenar una URL a un sitio web, genera una key SDM, que envía al backend antes de conectarse.

Esta key esta protegida añadiendo ya sea con el ID de la tarjeta, o con alguna key volátil como CMAC. Este backend puede asegurarse de que el mensaje es autentico y si fue encriptado de la forma correcta.

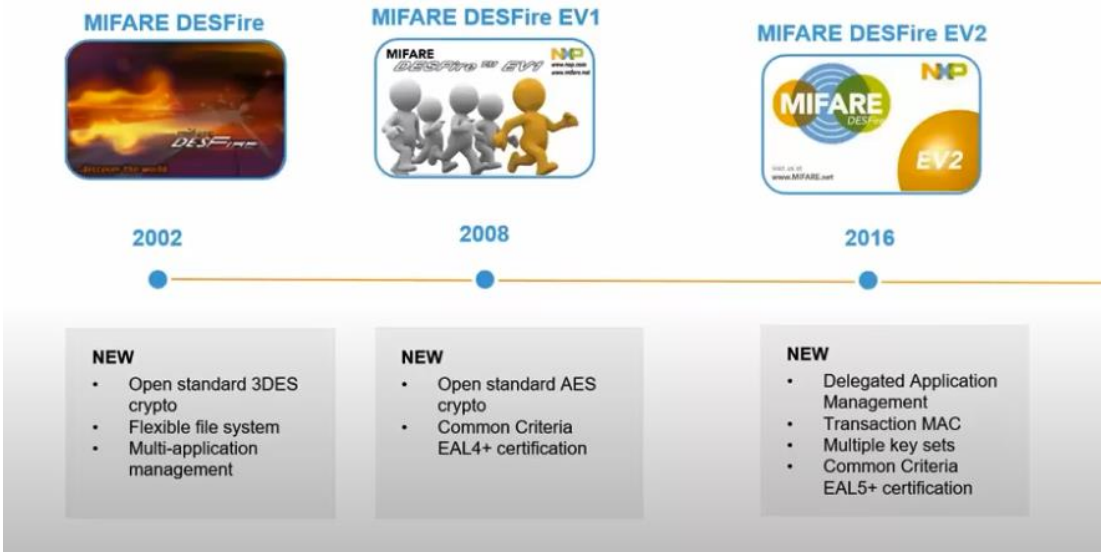


Figura 51 Evolucion de MiFare DesFire

IMXRT1050-EVKB, Como placa de desarrollo base

El IMXRT1050 es una implementación por parte de NXP del core de ARM-Cortex-M7. Que opera hasta 600Mhz lo que permite respuestas a tiempo real y eficiencia en la CPU. Permite muchísimas interfaces de conexión, como ethernet, GPS, todo tipo de displays, Bluetooth, interfaz para cámaras y interfaces analógicas.

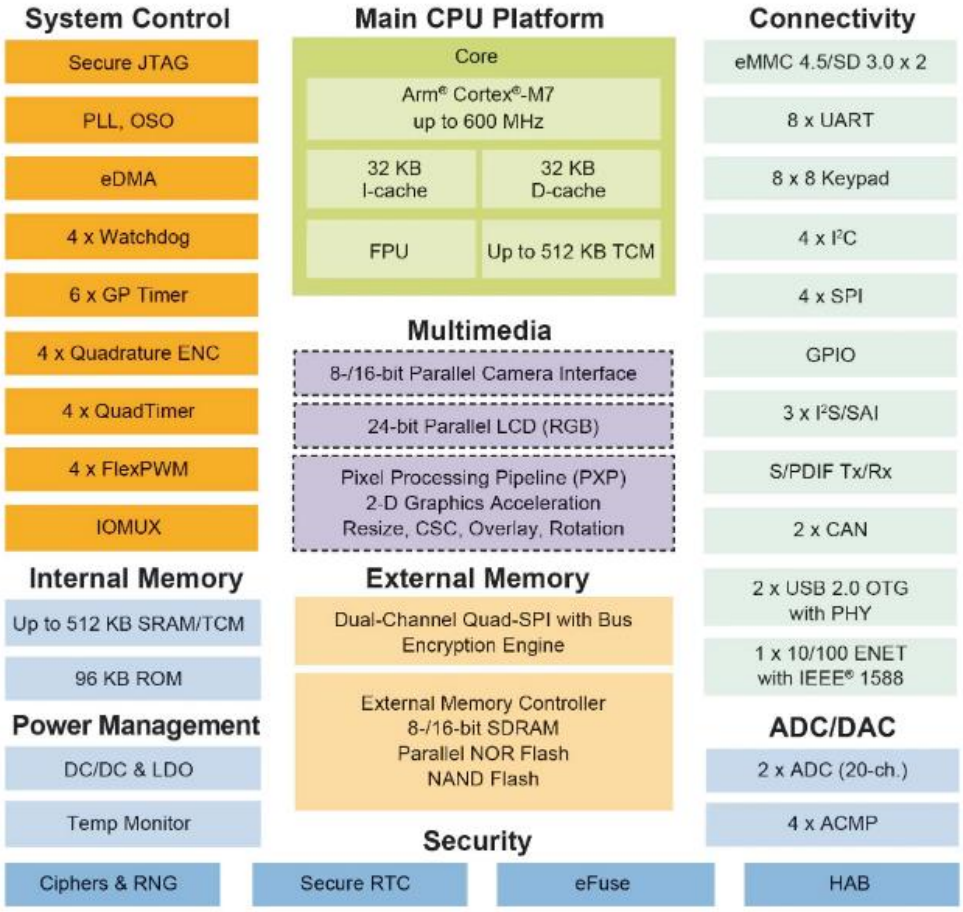


Figura 52 Diagrama de la board del IMXRT1050

Figura 53 IMXRT1050 Development Board

El IMXRT1050 es útil específicamente en aplicaciones destinadas al IoT, Interfaces hombre-maquina y control de motores.

Dentro de las especificaciones de seguridad tenemos.

Co-procesador criptografico, que permite AES-128 y ECC, y SHA-256.

Un motor de encriptacion de bus, que permite AES-128 y des-encriptacion “instantánea” (on-the-fly)

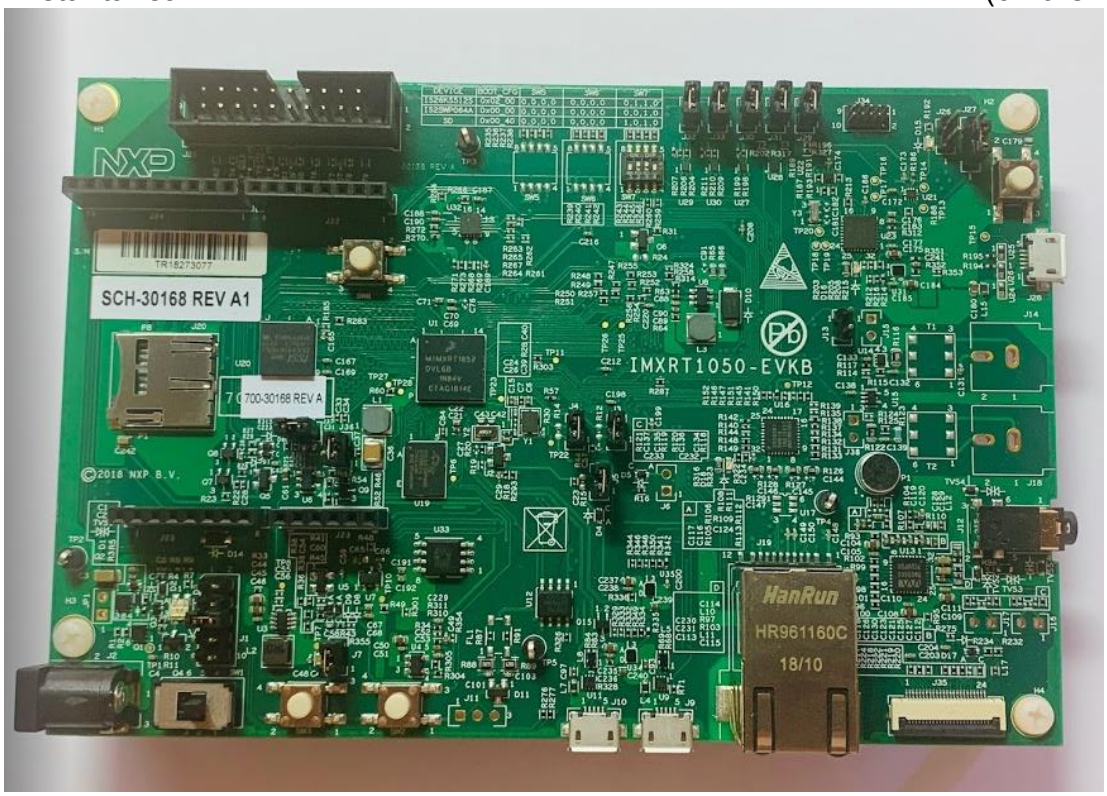


Figura 54 IMXRT1050 Board

Código y Librerías usadas.

Para poder explicar este proyecto de forma más sencilla, se separa en dos partes distintas. Una parte en la que explica cómo se realiza un descubrimiento básico de NFC, sin utilizar el SE050, y tan solo leyendo la UID de una tarjeta presente.

Programación de un Loop de Descubrimiento Básico

Basic Discovery es el uso de la librería NFC que provee NXP, para realizar un loop de descubrimiento básico de NFC. Se utilizara el IMX RT 1050 como Microcontrolador Host, y la board BLE-NFC-V2 como transciever NFC.

Para conectarlas, simplemente colocamos una encima de la otra, siguiendo este esquema de pines de comunicación SPI.

La librería NFC es un conjunto de software utilizado para desarrollar aplicaciones contactless con las aplicaciones y frontend de nfc. Estas librerías ofrecen las

operaciones mas comunes como leer, escribir, intercambiar etc, con todos los protocolos y estándares implementados. Además, tiene las librerías repartidas por capas. Primero por demos, luego por Sets de comandos y por último protocolos y Hardware.

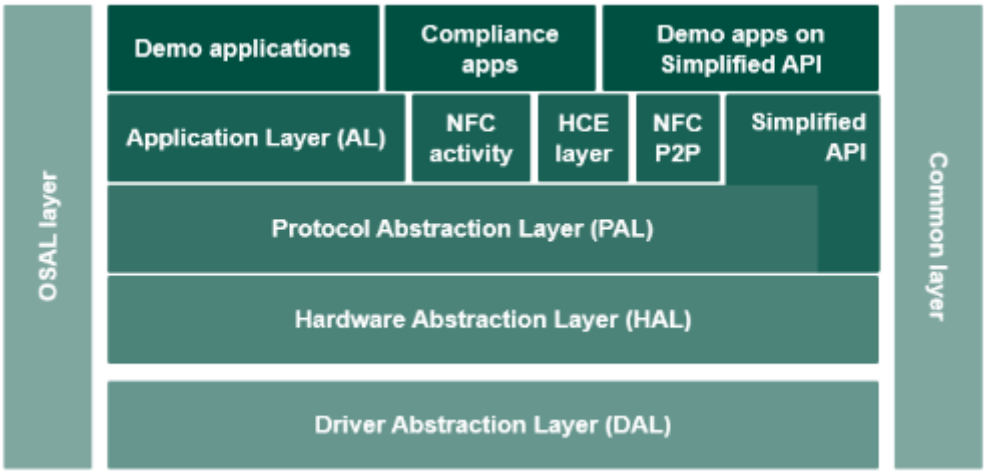


Figura 55 Esquema de la libreria NFC LIB y como esta separada por capas

Entre los demos se encuentran varios útiles para el desarrollo de aplicaciones con distintas tarjetas.

El IDE de programación que se ha utilizado es MCUXpresso, el programa que ejecutaremos en la librería es NfcrdlibEx1_BasicDiscoveryLoop. Este programa es genérico para SPI, así que hay que realizar una configuración de pines.

El RC663, el IC que se comunicara con IMXRT1050, lo realizara mediante SPI.

Realizamos la configuración de puertos para que puedan comunicarse. Para la comunicación SPI hace falta configurar, GPIO y los pines relativos a GPIO, el clock y el modo de SPI. Además, necesitaremos los drivers SPI.h.

Dentro de los ejemplos de NXP NFC se encuentran demos de configuración de SPI, y en documentaciones antiguas de otras boards de desarrollo, como la K64F de Freescale, contaban con demos que incluían la configuración SPI para comunicarse con CLR663. El código se encuentra en anexos [ANEXO]

Esta documentación de NXP y el demo de Discovery loop, no se encuentran configurados para IMXRT1050, ya que se trata de una board de 2019 y algunas demos no se han actualizado.

La configuración es sencilla, y se encuentra explicado en el Pliego de condiciones.

Las librerías de NFC trabajan con parámetros. El bloque main sigue el flowchart que se encuentra en planos.

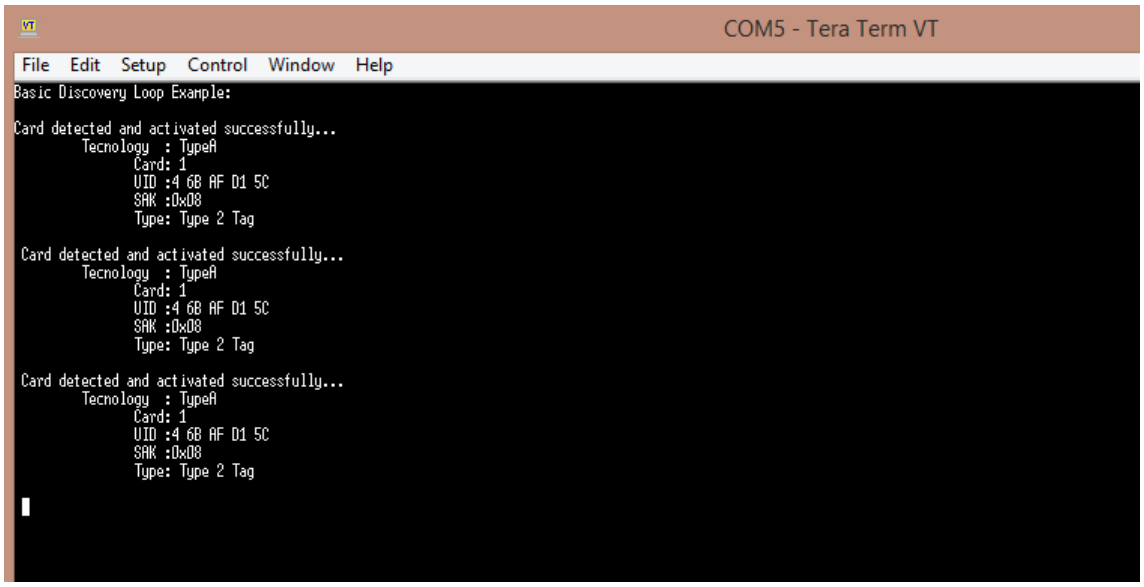


Figura 56 Serial Output de Tera Term en basic loop

Programación de un sistema de autentificación NFC con el SE050 como elemento seguro.

El Se050, en esta aplicación se utiliza para almacenar la clave secreta que se utiliza para establecer un canal seguro con una credencial MiFare, ya que el SE050 soporta las funciones de derivación autentificación y generación de claves de sesión de MiFare.

El sistema de comunicación es el siguiente. El SE050 se comunicara con I2C con la placa IMXRT. El CRC663, que se encuentra embebido en la board NFC-BLE V2, se comunica mediante SPI con el Micrcontrolador.

El caso práctico consta de 3 partes o pasos. La preparación del SE050, que genera una clave privadas AES a partir de números Pseudo-aleatorios.

Si se tratara de un modelo comercial, estas claves AES se generarían en un Módulo de seguridad Hardware. Estos módulos almacenan y generan claves criptográficas y se utilizan en líneas de producción. En esta línea de producción en concreto, almacena la raíz de la llave que se utiliza.

En nuestro ejemplo, una llave de sesión AES se genera en el SE050 que se enviara a IMXRT1050, y este controlara la comunicación mandado comandos NFC, con el CLRC663.

Preparar la Tarjeta DESFire, que igualmente lo hacemos con el SE050 en esta aplicación, pero un modelo comercial, estas tarjetas recibirían sus claves a partir de una línea de producción con un Módulo de Seguridad Hardware.

En este caso, el SE050 realiza una personalización de la tarjeta, y desencripta los datos en directo y además muestra cuales eran las llaves de sesión antiguas que se habían utilizado.

Realizar una autentificación simétrica utilizando las claves almacenadas en el elemento seguro SE050.

Para la programación de este proyecto, se utilizan las librerías NFC explicadas en el caso anterior, y el EdgeLockSE050 Plug&Trust MiddleWare [ANEXO]

La generación de la clave AES se genera en la librería ex_prepare_Se050.c , En esta se generan dos sets de claves que se almacenan en el SE050, y que se utilizaran para los siguientes pasos

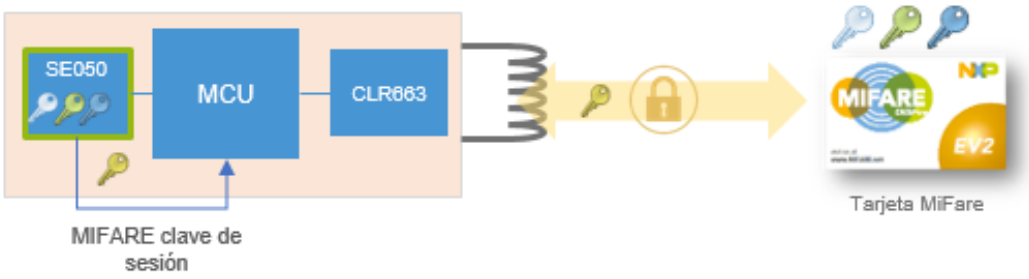


Figura 57 Esquema claves con SE050

Para la explicación resumiré en que consisten estas 3 funciones principales que ejecuta este programa. PrepareSE_050, PrepareCard y Auth, si esta última función Auth genera un canal seguro. La comunicación con SE050 se realiza siguiendo comandos APDU si nos queremos comunicar con las Applet o con T=1 I2C (Nuevo protocolo de I2C).

PrepareSE_050.

Mantiene siempre un constante status. Establecemos un enviroment para las claves de SCP03 Keys. La función es sss_key_store. Con esta función genera una KEY AES dinámica. Los #define generaran una clave AES para comunicarse con Mifare. Inicializa, localiza y prepara 2 claves AES.

El SE050 tiene dos claves internas que no dependen del programador, ya que son internas del SE050. Estas claves AES son propias de una aplicación y de una tarjeta. Existen muchas formas de almacenar y general claves y objetos en el SE050. Pero para moverlos a nuestro microcontrolador, que podría estar expuesto a eavesdropping, hay que seguir un protocolo de canal seguro.

Todos las claves y archivos almacenados en el SE050 se serializan siguiendo un cifrado, y solo se exportan objetos cifrados.

Para importar objetos seguros, el SE genera una clave publica que importa al microcontrolador. El “usuario” realiza el input y el tipo de SCP, junto con un. El applet del SE050 calcularа la clave privada a partir del secreto común y la clave publica, y el microcontrolador hará lo mismo. Al establecer el canal seguro ya se puede escribir en el SE. Se05x_API_ImportObject()

Estas claves se escriben desde el microcontrolador. Si queremos la máxima seguridad en nuestro prototipo, escribir las claves AES en los objetos mediante

[un HSM, o programar un microcontrolador para ejecutar SE_050, programarle nuestras claves, y usar otro microcontrolador o formatearlo, ya que estas claves nunca deberían de estar en plain text.

Este ejemplo las claves AES son oldkey[16] 0x0, y newkey[16]0x1000…

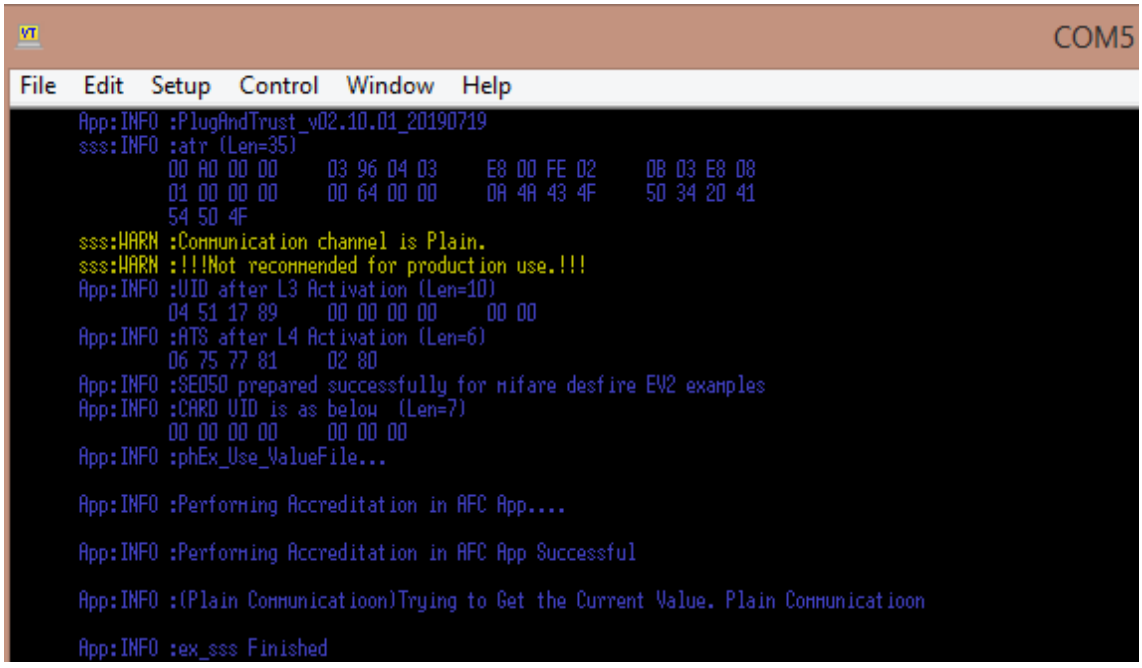


Figura 58 Tera Term tras la configuracion del SE050

PrepareMifareDFEV2

Después hay que personalizar la MiFare DesFire EV2. En este ejemplo se formateará en la tarjeta, la aplicación a llamar y las claves para comunicarse.

Para prepararla primero introducimos la clave AES que vamos a preparar. Esta aplicación Formateara la tarjeta, borrara todas las claves y aplicaciones que haya. Para que pueda hacer esto, se ha de conocer la clave AES “principal” . En este caso al ser tarjetas vírgenes podemos acceder a las zonas de memoria y sobreescribirlas. Al sobreescribirla, la clave AES de la tarjeta MifareEV2, coincide con la clave AES almacenada en el SE050 como clave para comunicar con la MiFareEV2.

Creeamos una aplicación. Esta aplicación basicamente son una serie de bytes que transmite la tarjeta cuando se le pregunta a que aplicación quiere acceder.

El lector le pregunta, la tarjeta responde aplicación 0x11 (AFC), y ambos se aseguran estar en el nivel de seguridad correspondiente, y iniciando un nuevo canal seguro en caso de no ser suficiente el nivel de seguridad.

```
App:INFO :PlugAndTrust_v02.10.01_20190719
sss:INFO :atr (Len=35)
         00 A0 00 00     03 96 04 03     E8 00 FE 02     0B 03 E8 08
         01 00 00 00     00 64 00 00     0A 4A 43 4F     50 34 20 41
         54 50 4F
sss:WARN :Communication channel is Plain.
sss:WARN :!!!Not recommended for production use.!!!
App:INFO :UID after L3 Activation (Len=10)
         08 90 FA C8     00 00 00 00     00 00
App:INFO :ATS after L4 Activation (Len=6)
         06 75 77 81     02 80
App:INFO :Performing Pre Personalization ......

App:INFO :Formating the card Successful

App:INFO :bCardUid[0] = 0x4

App:INFO :bCardUid[1] = 0x9

App:INFO :bCardUid[2] = 0x68

App:INFO :bCardUid[3] = 0x72

App:INFO :bCardUid[4] = 0xd5

App:INFO :bCardUid[5] = 0x63

App:INFO :bCardUid[6] = 0x80

App:INFO :bCardUid[7] = 0x0

App:INFO :bCardUid[8] = 0x0

App:INFO :bCardUid[9] = 0x0

App:INFO :Create AFC Application Successful

App:INFO :Select the AFC Application Successful

App:INFO :phEx_Create_ValueFile...

App:INFO : create Transaction MAC File  Successful

App:INFO : create Value File  Successful

App:INFO :***** Creating standard data file SUCCESS!!*******

App:INFO :ex_sss Finished
```

Figura 59 Terminal de tras la configuracion de PrepareMifareDFEV2

Autentication Mifare Desfire.

Para que ocurra el sistema de autentificación primero han de realizarse los dos programas anteriores y que la claves oldkey y btkey sean la misma. Si esto ocurre, el proceso se llama acreditación offline, que en caso de lograr una comunicación segura, que se cumple ya que las claves inyectadas en los anteriores pasos son la misma para tarjeta y SE050. El proceso que sigue para autentificarse es el siguiente:

Se ejecutan varias acciones, la tarjeta realizara su derivación de llaves a partir de la AES128.

El SE050 almacenará las llaves AES128 Maestras (0x0), y generará/derivará claves de sesión, que se las transmitirá al IMXRT1050. Además, también almacena un byte que indica el nivel de seguridad de la comunicación si se logra la autentificación.

También soporta funciones para cambiar entre claves de distintos “llaveros” almacenados en SE050. Esto depende 100% de la aplicación y dependerán del uso y del caso. Si la llamásemos, cambiaria oldkey por newkey.

Después el SE050 generará un nounce de 16Bytes y lo mandará cifrado con la clave oldkey, este nouce ira acompañado de identificador de llave, de 4 bytes.

El valor de autentificaciónNon se mantendrá en 0, hasta que no le llegue un valor igual al nounce sin decodificar.

También realiza una lectura del nounce codificado que le manda la llave, para decodificarlo.

Si la autentificación se completa, el valor del canal seguro dependerá de que aplicación puede acceder. Se mantendrá la comunicación cifrada en todo momento con la clave de sesión, que se mantiene almacenada en el SE050 que realiza todas las operaciones criptográficas de la comunicación, y manda el texto cifrado a él microcontrolador.

El objetivo del elemento seguro es que, en todo momento, el microcontrolador no pueda conseguir ninguna de las claves que tenemos almacenadas, ya sea en la tarjeta o en el SE050. Para ello hay que revisar las aplicaciones que programamos en él con mucho cuidado. En un uso práctico, estas claves se introducirían usando otros microcontroladores de los cuales se tiene la certeza que no han sido vulnerados. En nuestro caso, se introducen directamente modificando el valor.

Si llamamos a las tres aplicaciones seguidas, el resultado que obtenemos por el puerto serie es el siguiente.

La aplicación a la que se le pide comunicación dentro de la tarjeta cuando realizamos una autentificación es AFC App, (Access and Fare Control).

```
App:INFO :PlugAndTrust_v02.10.01_20190719
      sss:INFO :atr (Len=35)
               00 A0 00 00     03 96 04 03     E8 00 FE 02     0B 03 E8 08
               01 00 00 00     00 64 00 00     0A 4A 43 4F     50 34 20 41
               54 50 4F
      sss:WARN :Communication channel is Plain.
      sss:WARN :!!!Not recommended for production use.!!!
      App:INFO :UID after L3 Activation (Len=10)
               08 D2 5E 4A     00 00 00 00     00 00
      App:INFO :ATS after L4 Activation (Len=6)
               06 75 77 81     02 80
      App:INFO :Select the AFC Application Successful

      App:INFO :attempting to authenticate with cardkey = 0 and SeOObj ID =
2103308288
      App:INFO :
 CARD =====> SE050   16-byte Ek(RndB)  = (Len=16)
               2A 74 AE 94     94 7F 39 45     A2 F7 A0 8E     2D 08 21 59
      App:INFO :
  CARD <======= SE050  E(Kx, RandA || RandB') =  (Len=32)
               3A E3 12 EE     0A 73 F4 8F     9F 9A 78 09     13 E2 BA 63
               E5 6F 83 D3     15 1C 5B B3     08 0F 00 A0     62 2C 54 85


App:INFO :
  CARD ======> SE050  32-byte E(Kx, TI||RndA'||PDCap2||PCDcap2) = (Len=32)
               0E 7A 1D DE     8C 8F D9 32     C6 CC 36 6E     A5 89 16 DD
               AF 13 5D 4F     81 33 EE 08     EA 84 A5 45     73 22 EF F8
      App:INFO :
  CARD <====== SE050  E(Kx, RandA || RandB') = (Len=12)
               00 00 00 00     00 00 00 00     00 00 00 00
      App:INFO :Dumped Session Key is (Len=16)
               A7 C3 B6 C3     F9 C2 72 76     AA FF E5 EF     6A AB BE 5E
      App:INFO :Dumped Session Mac is (Len=16)
               97 82 3C 9C     E5 79 20 DE     B7 91 04 9B     E5 F9 6A 69
      App:INFO :Dumped TI is (Len=4)
               6E 1F 15 35
      App:INFO :pDataParams->wCmdCtr=0
      App:INFO : EV2 First Authenticate  Successful

      App:INFO :
 CARD =====> SE050   16-byte Ek(RndB)  =
 (Len=16)
               99 A8 0B 53     FD 81 B0 A0     62 8A 8D D7     09 ED 35 3C
      App:INFO :
  CARD <====== SE050  E(Kx, RandA || RandB') =
 (Len=32)
               31 29 79 24     DD 7E 57 DE     A6 D0 C5 A5     FC 47 4D E4
               46 A3 A9 6D     3F 5E C5 0D     B1 6F BC AF     A1 62 71 C5
      App:INFO :
  CARD ======> SE050  32-byte E(Kx, TI||RndA'||PDCap2||PCDcap2) =
 (Len=16)
               3F E7 46 48     BD C5 04 99     10 A3 AC C5     24 53 B4 46
      App:INFO :Dumped Session Key is (Len=16)
               2C D9 BD 43     EF 45 77 16     FD 7D BF 6F     9A 48 90 92
      App:INFO :Dumped Session Mac is (Len=16)
               00 1D 2D D9     72 41 85 52     00 72 7C C9     2D CD 91 77
      App:INFO :Dumped TI is (Len=4)
               6E 1F 15 35
      App:INFO :pDataParams->wCmdCtr=0
```

Figura 60 Terminal Taraterm tras una autentificacion correcta

Mejoras y accesorios al sistema: Cerradura On-line mediante AWS IoT

En este caso, se estudiará una mejora al sistema que no se ha implementado, pero que el prototipo tiene las librerías y hardware necesarias para implementarse

Figura 61 Logo de AWS

Si tenemos un edificio en el que hay personal que entra con muy poca frecuencia, ya sea un revisor de extintores o un trabajador de una empresa de mensajes, no nos interesa darle una llave que pueda abrir nuestro sistema siempre que quiera.

La solución mas sencilla a eso es ofrecer a través de un servidor, la clave privada AES de autentificación, pero el sistema de comunicación ha de estar encriptado y ser seguro. Las comunicaciones si no siguen un protocolo no son seguras, y nuestro objetivo con este proyecto es conseguir el máximo posible de seguridad

La manera en la que los ordenadores se comunican con servidores web y entre si de manera segura es mediante certificados CA. Un certificado CA requiere que un dispositivo este registrado en una CA o Autoridad Certificadora. Esta CA se encarga de almacenar las claves publicas de los dispositivos a los que ha certificado, y además les extiende un certificado de verificación.+

El sistema es un intercambio público del tipo Diffie-Hellman, pero en el que los dos dispositivos no se comunican entre si mandándose claves públicas, si no que se mandan certificados que redireccionan a el servidor de la CA, que devuelve la clave pública del dispositivo que ha extendido el certificado. Este certificado lo que hace es asegurar que el dispositivo con el que te comunicas es quien dice ser, ya que el inicio de la cadena de seguridad está en el CA.

En este ejemplo, la nube es el Nucleo IoT de Amazon Web Service, ya que su implementación es sencilla y es gratuito para prototipos. De ahora en adelante AWS Core, sirve como CA, y el dispositivo que se registre recibirá un certificado de verificación de AWS, en este caso el SE050.

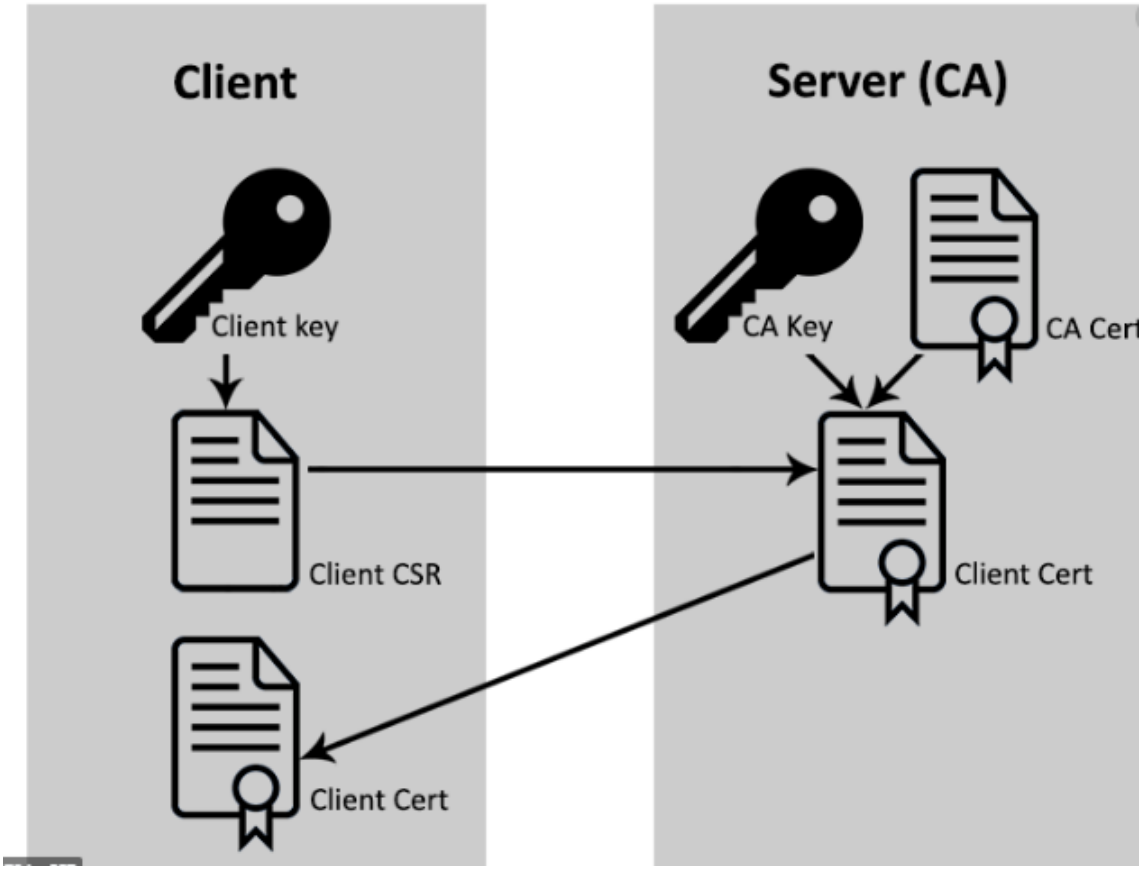


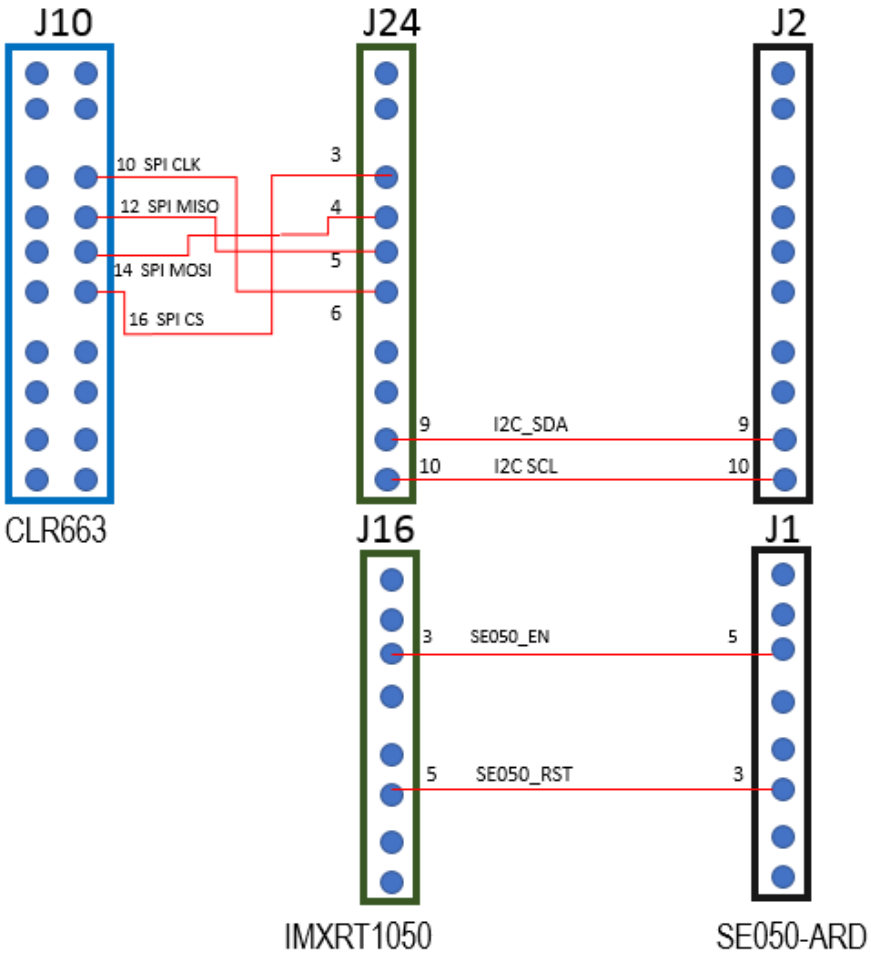
Figura 62 Sistema creación de certificados.

Para poder realizar esto, primero es necesario inyectar la clave publica de AWS en el SE050, junto con el certificado de esa misma clave pública. De esta forma el SE050 puede enviar información segura a AWS, sabiendo que se está comunicando con él. Ahora el SE050 generará un par de claves, y mandará la pública junto con su ID y el código de usuario utilizado el certificado, y AWS responderá enviando un certificado de verificación. Para comunicarse, ahora ambos utilizan sus certificados y pares de llaves, estableciendo una comunicación segura mediante internet.

Ahora, desde la interfaz de AWS puedes comunicarte de manera segura con el SE050, sabiendo que toda la comunicación esta encriptada. Desde esta interfaz puedes inyectarle al SE050 claves de autentificación y retirarlas. Puedes enviar usando la interfaz de AWS a un trabajador la clave AES que acabas de inyectar, y retirarla cuando desees.

Además, dado que el SE050 admite distintas aplicaciones con distintas claves, puedes hacer desde el origen que solo una de ellas pueda comunicarse con AWS. Así, si crees que te han podido robar la clave de acceso a AWS, bastaría con hacer un RollKey o cambio de llavero en esa aplicación, sin necesidad de conectarse a AWS para quitarle los permisos.

3 Planos



	FECHA	NOMBRE
Dibujado	02/09/2019	David Gonzalez
Comprob.	02/09/2019	David Gonzalez
D.S.Normas	U.N.E	Tolerancia General

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL

Grado en Ingeniería Electrónica Industrial

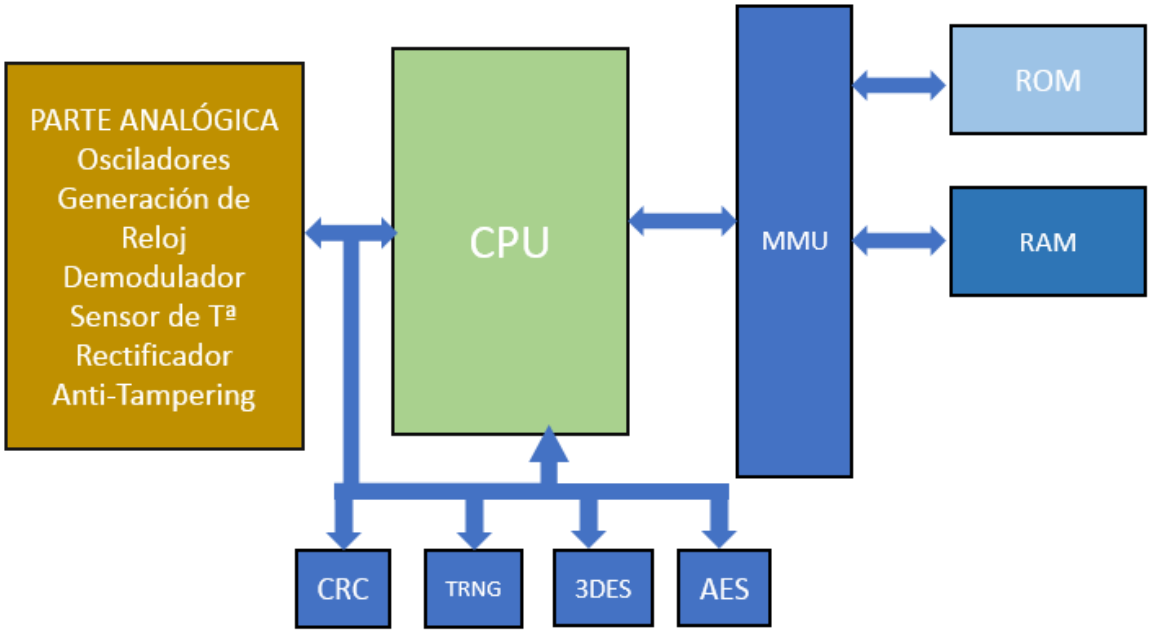
Escala S/E

proyección

PLANOS TFG

NÚMERO: 1

Diagrama de Conexiones del TFG



	FECHA	NOMBRE
Dibujado	02/09/2019	David Gonzalez
Comprob.	02/09/2019	David Gonzalez
D.S.Normas	U.N.E	Tolerancia General

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL

Grado en Ingeniería Electrónica Industrial

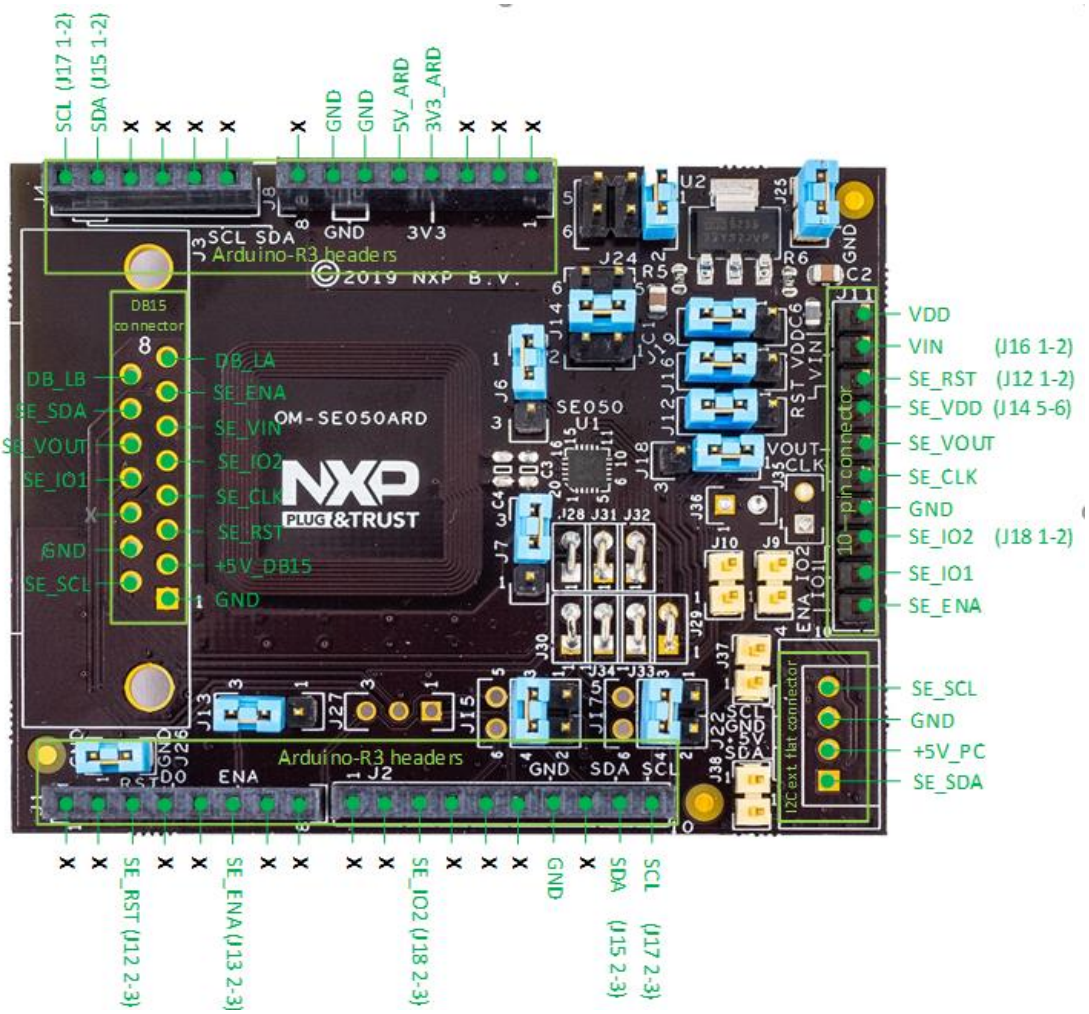
Escala S/E

proyección

PLANOS TFG

NÚMERO: 2

Esquemático de una tarjeta MiFare



	FECHA	NOMBRE
Dibujado	02/09/2019	David Gonzalez
Comprob.	02/09/2019	David Gonzalez
D.S.Normas	U.N.E	Tolerancia General

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL

Grado en Ingeniería Electrónica Industrial

Escala S/E

proyección

PLANOS TFG

NÚMERO: 3

Diagrama de Conexiones del SE050

4 Pliego de Condiciones

Prototipo Básico

El prototipo básico, se utiliza como ejemplo del uso de la librería NXP, para reducir algo de carga teórica al prototipo completo, que es este prototipo pero con el Secure Element SE050 ya unido.

Software y Librerías.

El software que se ha utilizado para programar es MCUXpresso IDE v.11. El idioma elegido para la programación ha sido C para sistema embebidos, pero tanto el IDE como el resto del proyecto aceptan otros lenguajes de programación.

Este IDE necesita de un SDK, Software Development Kit especifico para la Placa IMXT1050RT. Este Software se puede descargar desde la web de NXP, donde se pueden personalizar las librerías que vienen unidas a esta descarga. La librería de NFC no se encuentra aquí.

Los drivers necesarios para este proyecto, especificados en el header del código son :

Fsl_clock , Fsl_common, fsl_gpio, fsl_iomux, fsl_lpuart,fsl,iomuxc . Estos drivers se encuentran divididos en archivos específicos para hardware .h y con funciones especificas .c, y ambos son necesarios para implementar

El Software que se ha utilizado para la programación y el ruteo de pines es MCUExpreso Config Tools V7.

La librería que se ha utilizado para programar la demo NFC Discovery_Loop_Basic es NxpNfcRdLib_RC633

Esta librería nos da phApp_Init. Y NfcrdlibEx_1 discovery loop.

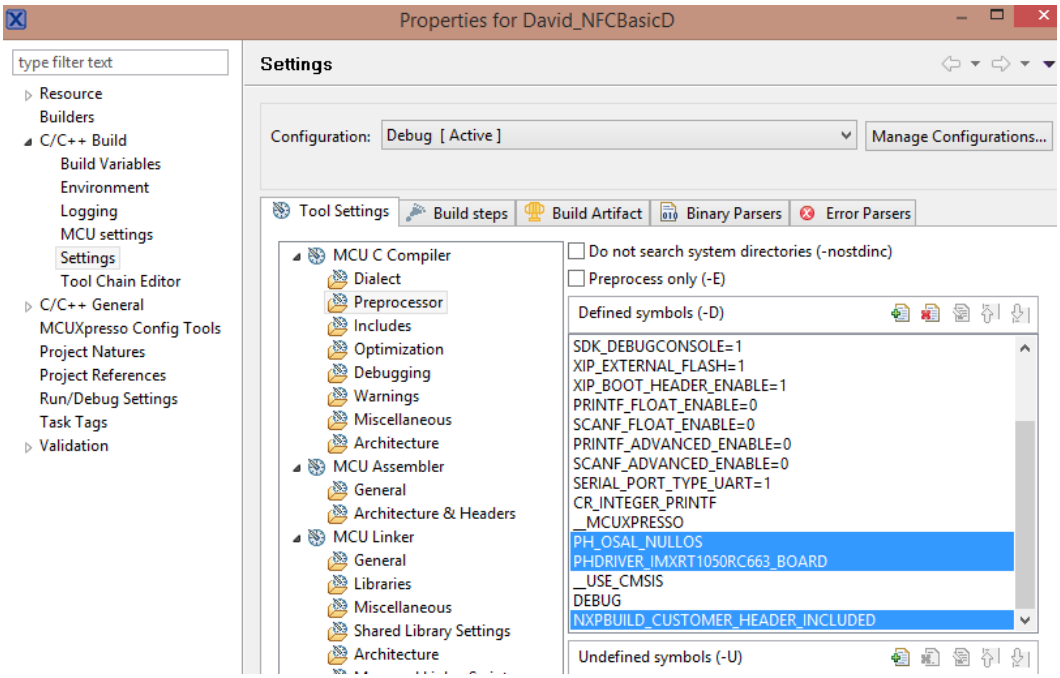
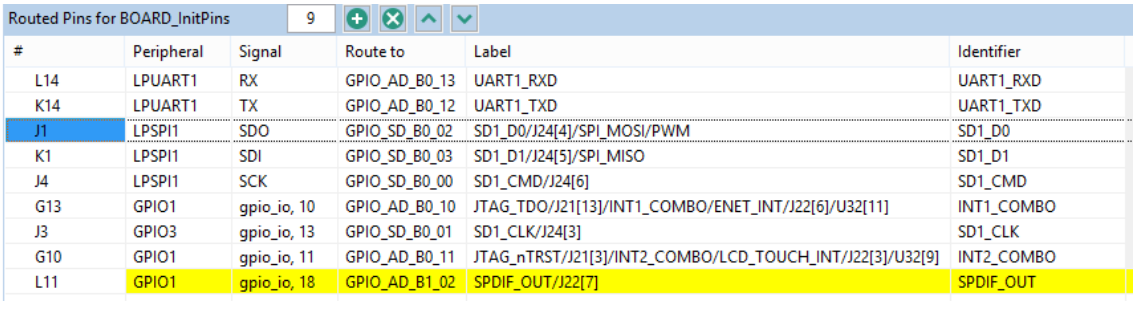


Figura 63 Librerías a añadir en el compilador

.

Al usar IMXRT1050, la librería de NFC todavía no lo tiene como opción en sus archivos, así que hay que modificar el archivo board del K82F, y sustituirlo por el archivo Board de IMXT1050RT, y modificar todos los llamamientos a este archivo. En Anexos, se puede ver la parte de código que se modifica para este proyecto.

Realizar el siguiente diagrama de pines en MCU Expreso_Config Tools, siguiendo el diagrama de conexión que hayamos especificado.



#	Peripheral	Signal	Route to	Label	Identifier
L14	LPUART1	RX	GPIO_AD_B0_13	UART1_RXD	UART1_RXD
K14	LPUART1	TX	GPIO_AD_B0_12	UART1_TXD	UART1_TXD
J1	LPSPI1	SDO	GPIO_SD_B0_02	SD1_D0/J24[4]/SPI_MOSI/PWM	SD1_D0
K1	LPSPI1	SDI	GPIO_SD_B0_03	SD1_D1/J24[5]/SPI_MISO	SD1_D1
J4	LPSPI1	SCK	GPIO_SD_B0_00	SD1_CMD/J24[6]	SD1_CMD
G13	GPIO1	gpio_io, 10	GPIO_AD_B0_10	JTAG_TDO/J21[13]/INT1_COMBO/ENET_INT/J22[6]/U32[11]	INT1_COMBO
J3	GPIO3	gpio_io, 13	GPIO_SD_B0_01	SD1_CLK/J24[3]	SD1_CLK
G10	GPIO1	gpio_io, 11	GPIO_AD_B0_11	JTAG_nTRST/J21[3]/INT2_COMBO/LCD_TOUCH_INT/J22[3]/U32[9]	INT2_COMBO
L11	GPIO1	gpio_io, 18	GPIO_AD_B1_02	SPDIF_OUT/J22[7]	SPDIF_OUT

Figura 64 Configuración de pines de CR663 con IMXRT

El Terminal Serie que se ha elegido para este proyecto es Tera Term, pero otras opciones validas también estudiadas y comprobadas que funcionan son el Terminal PuTty y el terminal del IDE de MCUXpresso.

Este terminal se comunica a 115200 baudios, 8 bits, 0 bit de paridad y 1 bit de Stop.

Diagrama de conexión

El diagrama de conexión que se ha seguido es el siguiente. Dentro de la documentación de IMXRT se puede encontrar todos los GPIO que permiten conexión SPI. La razón por la que en este proyecto se usan estos es por comodidad, ya que la Board BLE-NFC de CR663 esta diseñada para implementarse como un Shield de Arduino, y los pinouts de IMXRT están distribuidos para poder permitir la conexión a un Arduino.

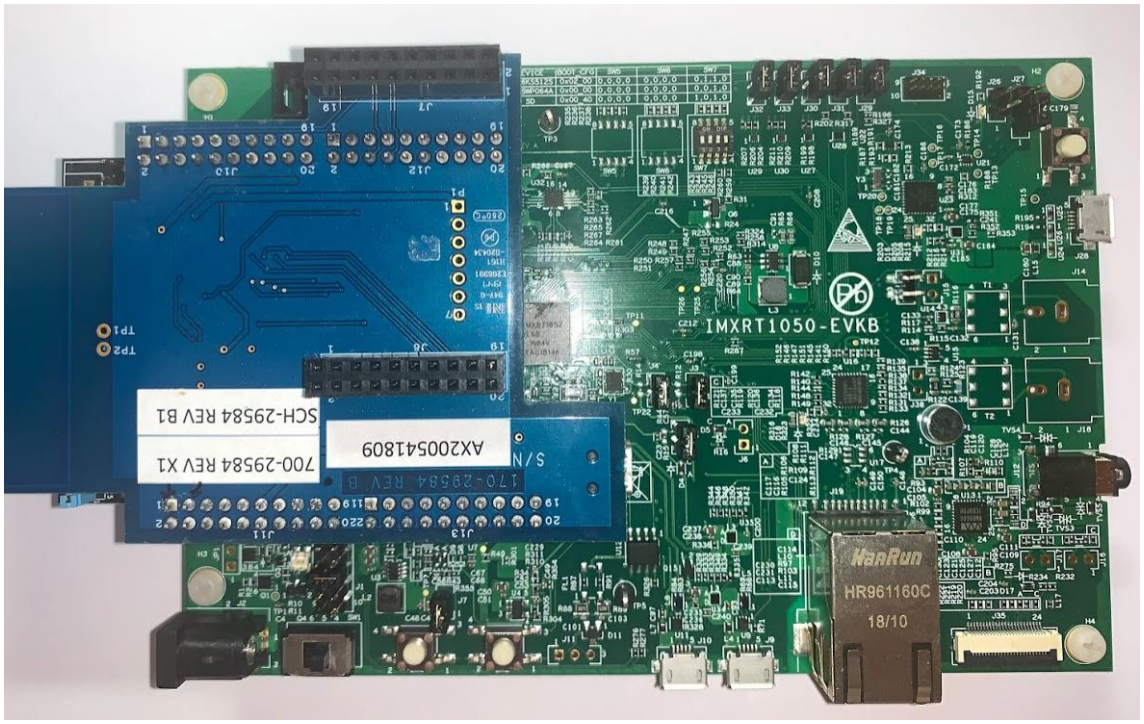


Figura 65 Montaje de RC663 con IMXRT Board

Esta conexión es SPI, así que es necesario soldar entre si las siguientes resistencias en la board IMXRT1050, según lo indicado por el fabricante.

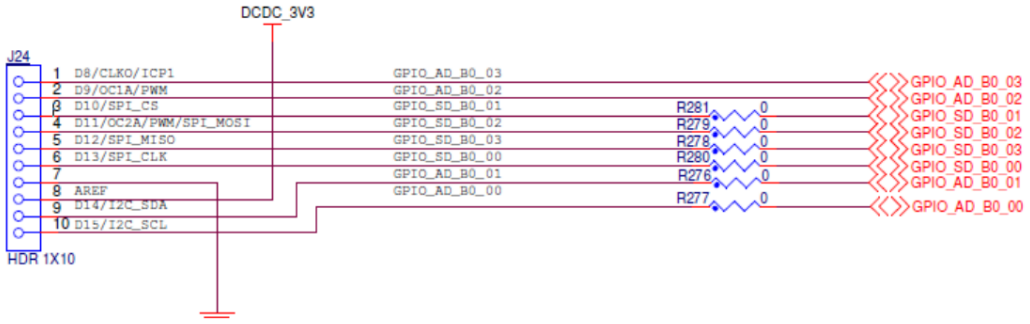


Figura 66 Esquemático de SPI en una IMXRT board nueva

Para asegurarse de que los pines SPI están bien soldados, ejecutar el demo de SPI incluido en la librería IMXRT1050, uniendo con cables los pines que se indican en el ejemplo.

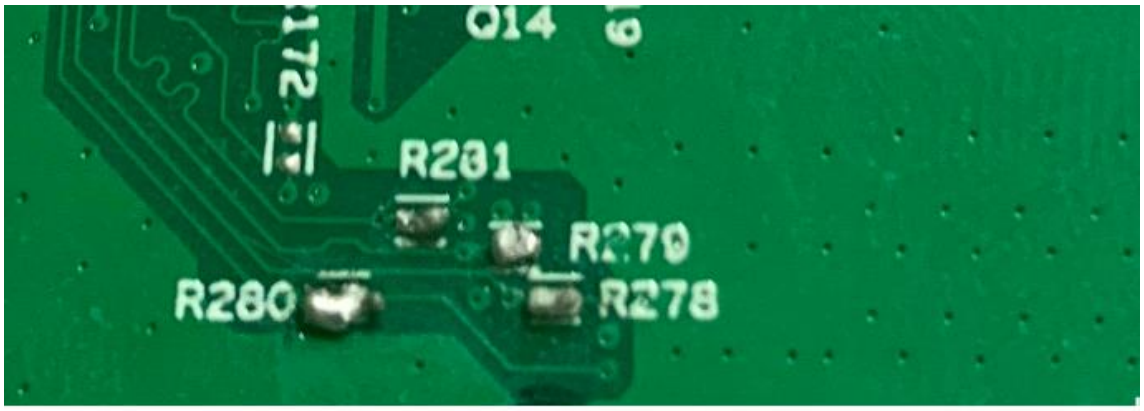


Figura 67 Soldaduras de resistencias de 0Ohms de para conexión SPI

La conexión con el ordenador se hará por el pin de alimentado y datos SDA de la board IMXRT1050.

	IMXRT	**BLE-NFC**
MOSI	J24-P4	J10-P14
MISO	J24-P5	J10-P12
SPI SCK	J24-P6	J10-P10
SPI CSEL1	J24-P3	J10-P16
RESET	J22-P6	J12-P6
IRQ	J22-P5	J12-P8
GND	J25-P4	J11-P11
3V3	J25-P6	J11-P15

Prototipo Completo

Este prototipo busca ser un ejemplo de un sistema de autentificación que contenga un elemento seguro para el almacenamiento de credenciales. En este caso, usaremos las mismas librerías que se encuentran explicadas en el prototipo básico, pero añadiendo el middleware de SE050. Este middleware recibe el nombre comercial de plug and trust.

Figura 68 - Montaje de SE050 y CLR663 sobre IMXRT1050

Los jumpers deberán de seguir este orden:

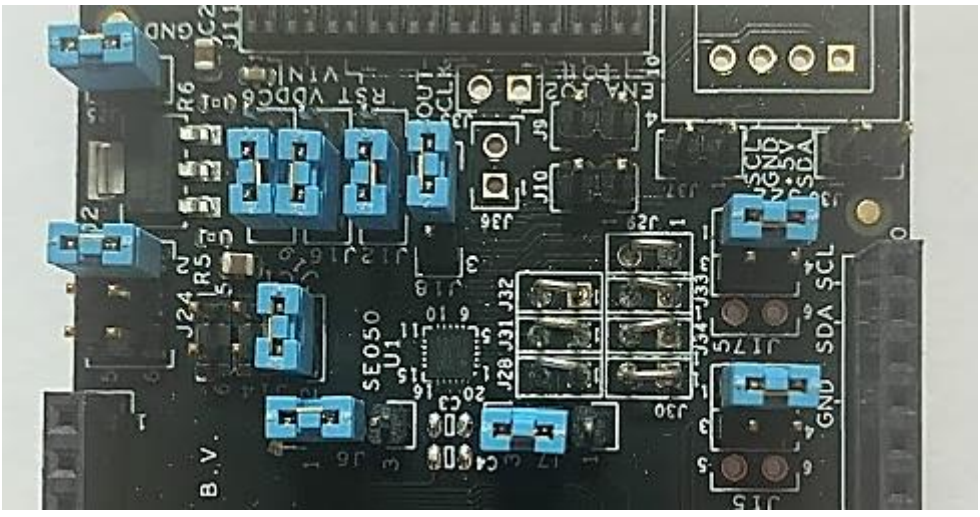


Figura 69- Posicionamiento de Jumpers del SE050

De estos jumpers el que nos afecta es J14, que hemos de modificar a la posición por defecto de 2-3 a 1-2 para que utilice el microusb como fuente de alimentación.

J15 y J17 hay que ponerlos en la posición 1-2, ya que son los que se encarga de establecer las salidas J2-5 y J2-5 Que son SDA y SCL de I2C

Software y Librerías.

Usamos las 3 funciones de NXP Library que se corresponden a NFX_MFDESFIREAV2.

Cuando se transmitan las llaves POLICY_MIFARE, el

OBJ_ALLOW_DESFIRE_AUTHENTICATION tiene que estar a 1.

En el código fuente se encuentra Ex_prepare_se05.c, que llama a las librerías de inicialización de apps de NXP, y a se05_MFDInit, de inicialización de Mifare.

El setup Inicial del SE050, InitialSetupSE050() lo realizamos nosotros como entidad segura, suponiendo que nuestro sistema no tiene ningún tipo de vulnerabilidad.

En este setup definimos los objetos seguros que va a almacenar el SE050. Esta comunicación se hace sin cifrar, desde microcontrolador al SE050. Importante, esto en caso de poderse evitar, (usando un HSM) se debería de hacer, ya que es la única vulnerabilidad actual del sistema.

En este ejemplo de autentificación, los objetos que definimos son las claves AES que vamos a inyectar en el SE050 esta inyección se hace con la función sss_key_object_init, con la clave que queremos inyectar, y que en la siguiente función este inyectara en la tarjeta MiFare Desfire.

```
sss_status_t InitialSetupSe050(sss_key_store_t *pkeyStore, uint32_t keyId, uint32_t derivedkeyId)
{

    sss_status_t sssStatus = kStatus_SSS_Success;

     uint8_t key[KEY_BIT_LEN / 8] = { 0 };
    sss_key_part_t keyPart = kSSS_KeyPart_Default;
    sss_cipher_type_t cipherType = kSSS_CipherType_AES;
    size_t keyByteLenMax = KEY_BIT_LEN / 8;
    sss_object_t oldKey;
    sss_object_t newKey;

        sssStatus = sss_key_object_init(&oldKey, pkeyStore);
        if (sssStatus != kStatus_SSS_Success) {
           LOG_E("Key Obj initialization failed");
            goto exit;
```

Figura 70 Inicialización/inyección de objetos tipo clave en el SE050

La configuración hay que inicializar, después realizar un sss_key_object_init, y después un sss_key_store_set . De esta manera, ya tenemos los objetos seguros almacenados en el SE050.

NewKey es un objeto definido, que, si una función de cambiar llaves ocurre, Newkey será la sustituta. En este caso la función es autentificar, así que una tarjeta MiFareDesFire deberá de autentificar contra NewKey.

Esta función añade seguridad ya que, si una tarjeta ha sido comprometida, no hay que injectar nuevas claves en las tarjetas, simplemente decirle a la aplicación que solicite a la tarjeta MiFare la llave número 2, NewKey de la aplicación.

Después queda configurar las llaves de cada aplicación y las aplicaciones a inyectar en el SE050, en caso de querer más aplicaciones (Que autentifique contra otros lectores, etc)

En el archivo de la librería SE_050 Key, están definidas todas las claves que el SE050 usa para las aplicaciones. Mac, Sesión, de Comunicación, etc. En un caso real, este archivo lo editaría un HSM, que es un diseño de hardware muy seguro, que almacena credenciales y llaves en un servidor. Este HSM es el que se encargaría de crear de manera aleatoria las llaves necesarias para la aplicación, y de comunicarse con el SE050 y de inyectarlas, durante tiempo de fabricación. Entre los objetos inyectados en el SE050 en fabricación se inyecta también un certificado de verificación, que permite una conexión segura entre dispositivos en internet, y posteriormente con esa inyección se pueden modificar las claves del SE050.

5 Presupuesto

Presupuesto del Prototipo

Descripción de unidades del prototipo

Nombre Identificactivo	Descripción
OM-SE050ARD Dev Kit	Placa de prototipo de Elemento Seguro SE050
i.MX RT1050 Evaluation Kit	Placa de prototipo de conexiones de microcontrolador IMX RT1050 (Cortex M7 de NXP)
BLE-NFC-V2	Placa de prototipo
Mifare Desfire EV2	Tarjeta NFC de prototipo

Descripción de Software de Prototipo

Nombre del Software	Descripción
MCUExpreso IDE	Software de programación de microcontroladores por NXP
SDK IMXRT1050	Software para la IDE de MCUExpreso con toda la layer del hardware necesario para controlar
MCU Config Tools	Software de programación de controladores de NXP
TeraTerm	Software open source de simulación de terminal Serie
RFIDiscover	Software de lectura y escritura de NDEF
Amazon IOT AWS	Servidor online donde monitorizar la cerradura
NFCNXPLIB	Librería de NFC de NXP
SE050 Plug&Trust MW	Librerías de gestión de comandos APDU para que un microcontrolador se

	comunique de manera segura con el SE050
AWS free RTOS	Librerías de gestión de certificados y comandos para comunicarse con AWS

En cuanto al gasto personal se reparte las horas dedicadas por el autor de este TFG a 4 personas ficticias encargadas de las labores detalladas en el diagrama a trabajadores ficticios.

El personal de redacción de documentos, se encargará de la redacción de documentos sobre la tecnología, la memoria y el producto. El tiempo empleado en estas labores en este TFG ha sido de 125h

Un programador/diseñador, ingeniero electrónico que estudie el prototipo, lo diseñe y estudie las librerías, las implemente y modifique según el objetivo. El tiempo empleado en estas labores en este TFG ha sido de 160h

Un programador senior, que se encargue de los arreglos de código, modificaciones al informe y coordinación en general. El tiempo empleado en estas labores en este TFG ha sido de 70h

Personal	Numero de h	Precio/h	Total
Redacción	125	16.00€	2000.00€
Programador	160	24.00€	3840.00€
Programador S	70	35.00€	2450.00€
Total			8290.00€

Precio de unidades del prototipo

Nombre Identificativo	Precio unitario
OM-SE050ARD Dev Kit	58€
i.MX RT1050 Evaluation Kit	79€
BLE-NFC-V2	50€
Mifare Desfire EV2	1.27€
Total (€)	188,27€

El Software utilizado es todo de licencia abierta, en el que las empresas no se hacen responsables de su mal uso o programación.

Amazon IOT tiene una versión gratuita para usuarios, y una versión premium para empresas que asciende a 100€ mensuales.

Presupuestos de casos comerciales

Presupuesto caso comercial pequeño, puertas Smart.

Primero se estudiara un caso comercial, en el que una empresa decide añadir a sus cerraduras, una cerradura inteligente unido con Amazon IOT, para que un cartero pueda acceder con su tarjeta si se le da permiso en la app de Amazon en un horario y día determinado. Estos chips y tarjetas se venden en lotes muy grandes, de 10000 Uds o más.

Suponemos que la empresa de puertas que quiera sacar un modelo de cerradura interligente, para empezar, 200 “cerraduras” y 1000 tarjetas, 5 por cerradura. El precio de creación y montaje de PCBs varía entre fabricantes, así que establecemos s el precio estándar para placas de cuatro caras más montaje.

No se incluye aquí precio de la cerradura en sí, ni del exterior ya que no forma parte del proyecto electrónico.

Nombre Identificativo	Precio Unitario
Fabricación de PCB	1,25€
Montaje de PCB + Resistencias, condensadores y jumpers SMT	4,15€
MIMXRT1052DVL	5,32€
CLRC66302	3,40€
SE050A1HQ1	1,07€
Total Unitario	15,19€
Tarjeta Mifare Desfire EV2 (1000 Uds + Programación, 0,80+0,12)	0,92€
Lector + 5 Tarjetas	19,79€
Total 200 Uds	3958€

Para este trabajo, suponemos que al ser tan pocas unidades el montaje de la PCB se hace en una empresa externa, y que estos realizan el montaje de la PCB y el montaje de las partes del lector. Quedando el precio de fabricación de cada lector en 15,19€ y 19,79€ con las llaves.

Estas llaves se han de programar y unir al SE050 de manera segura, así que solicitaremos el alquiler de un HSM a la entidad que nos diseñe los chips. No he encontrado precios online, así que he añadido 12 céntimos por unidad en inserción segura de claves.

En cuanto a los trabajadores, un programador que explique el sistema, lo diseñe según las necesidades de la empresa y modifique según el sistema.

Un diseñador, que se encargue de la tecnología de la PCB, encargo y selección de la empresa que las montará. Diseñara la antena y optimizara el producto.

Un programador senior, o supervisor, que se encargue de correcciones y modificaciones.

Personal	Numero de h	Precio/h	Total
Programador	100h	24.00€	2400.00€
Diseñador	200h	24.00€	4800.00€
Programador S	100h	35.00€	3500.00€
Total			13100.00€

Apartado	Precio Total
Fabricación	3950€
Diseño y Programación	10700€
Total	14658€
Total por unidad Construida	73.29€

Los beneficios de este proyecto serán o muy pequeños, o nulos, ya que los gastos de programación son demasiado elevados para tan pocas unidades, y lo más rentable seria pedir la tecnología NFC a otra empresa.

Aunque este tipo de tecnologías están ya a la venta, y el precio de la cerradura comercial estándar cuesta 20€, si le añadimos el motor que controle la cerradura el precio de cada unidad ronda los 100€.

Este precio es factible, ya que la mayoría de las cerraduras de este estilo se encuentran entre los 180-350€, y podría estudiarse como una prueba de mercado o modernización de una empresa que se dedica a la manufactura de cerraduras de puertas.

Presupuesto caso comercial grande, Transporte de una Comunidad.

En este supuesto una comunidad ha decidido basarse en el actual contrato del consorcio de transportes de la comunidad de Madrid. Este proyecto se realizo en 2018 desde el Area de Ingeniería en el proyecto estación 4.0. Por lectores y tecnologías de Mifare Desfire.

La comunidad de Madrid decidió en el 2018 solicitar el suministro de 50000 Tarjetas NFC, como la que se ha visto en apartados anteriores, del tipo MiFare DesFire EV1

Reinstalar 1600 lectores RFID, en autobuses estaciones de metro y sistemas de pago de estaciones de metro, y 50000 tarjetas. Este contrato buscaban dividirlo

entre varias empresas para poder solicitar tarjetas en caso de necesidad, aunque al final fue Telefónica SAU quien licitó los 4 contratos que el gobierno de Madrid saco a adjudicación pública.

El presupuesto de Hardware es el siguiente, y los precios son derivados desde mouser para 1600 uds y 50000, y se ha decidido reducir el precio un 10% debido a que se trata de una gran empresa que realiza numerosos pedidos.

Presupuesto para la electrónica interna del lector. (Intentará que el diseño aproveche al máximo la electrónica que ya estaba instalada, de esta manera alimentación y sistema de motores de las puertas no se modifica, solo el lector de tarjetas)

Nombre Identificativo	Precio Unitario
Grabado PCB y Montaje	0.50€
Componentes SMD 30uds (R / C)	0.03€
MIMXRT1052DVL	3,32€
CLRC66302	2,40€
SE050A1HQ1	0,83€
Antena	0.05€
IC de alimentación(33YSQWDP)	0,50€
IC de conexión Ethernet	0.9€
Precio componentes de unidad de Lector	8,53€
Total 1600 Uds	13648.00€

El presupuesto de las tarjetas

Nombre	Precio Unitario
IC Mifare Desfire EV2	0,52€
Antena	0.05€
Vinilado de Tarjeta	0.05€
Total Tarjeta unitario	0.62€
Total 50000 Tarjetas	31000€

El número de horas desglosado por las distintas tareas que se invierten en el proyecto.

El proyectista, que diseñara todo el proyecto, organizara y será el responsable legal de que se cumpla el pliego de condiciones del proyecto

El instructor/coordinador, que se encargara de formar siguiendo las directrices de este proyecto a los ingenieros de mantenimiento instalación y de bases de datos.

Los programadores, que se encargaran de configurar el código de los lectores para las aplicaciones que le correspondan a los lectores y tarjetas, además de crear aplicaciones móviles y sistemas seguros en la nube.

Los ingenieros de bases de datos y del Hardware security Module. Se encargaran de establecer el sistema que se comunique de manera segura tanto con los SE050 como con las tarjetas Mifare para el control de credenciales. También se encargaran de las bases de datos personales online, pagos

Personal	Numero de h	Precio/h	Total
Proyectista	300h	50.00€	15000.00€
Instructor	200h	35.00€	7000.00€
Programador	300h	24.00€	7200.00€
Diseñador de PCB	200h	24.00€	4800.00€
Ingenieros de Instalación y Mantenimiento	1600h	20.00€	32000€
Ingenieros de Bases de Datos y HSM	500h	24.00€	12000€
Total			78000.00€

Apartado	Precio Total
Precio Lectores	13648.00
Precio Tarjeta	31000.00€
Mano de Obra	78000.00€
Total	122648.00€

6 Anexos

Anexo Isos

14.2.2 List of Standards

26. BImSchV: 'Sechsundzwanzigste Verordnung zur Durchf¨uhrung des Bundes-Immissionsschutzgesetzes – Verordnung ¨uber elektromagnetische Felder', with explanatory section, in Wolfgang Kemmer, '*Die neue Elektrosmog-Verordnung*', H. Hoffmann GmbH Verlag, Berlin, 1997, ISBN 3-87344-103-9
AIAG ARF-1 Application Standard for RFID Devices in the Automotive Industry
AIAG B-11 Tire and Wheel Identification Label Standard
ANSI/INCITS 256 Radio Frequency Identification (RFID), NCITS 256 defines a standard for Radio Frequency Identification (RFID) for use in item management. This standard is intended to allow for compatibility and to encourage interoperability of products for the growing RFID market in the United States
ANS/INCITS 371: Information Technology – Real Time Locating Systems (RTLS).
Part-1: 2.4 GHz Air Interface Protocol
Part-2: 433 MHz Air Interface Protocol
Part-3: Application Programming Interface
ANSI/MH 10.8.4 RFID for Returnable Containers.
AWWA IMT61457 The Use of Mobile and RFID Data and Field Force Integration in a Major Water Utility
CEPT T/R 60-01: Low-power radiolocation equipment for detecting movement and for alert (*EAS*). Technical Recommendation. http://www.ero.dk
CEPT T/R 22-04: *Harmonisation of frequency bands for Road Transport Information Systems (RTI) (toll systems, freight identification)*. Technical Recommendation. http://www.ero.dk

ECMA-340: see ISO/IEC 18092 (NFCIP-1)
ECMA-352: see ISO/IEC 21481 (NFCIP-2)
ECMA-356: see ISO/IEC 22536 (NFCIP-1; RF Interface Test Methods)
ECMA-362: see ISO/IEC 23917 (NFCIP-2; Protocol Test Methods for NFC)
EN 50061: *Safety of implantable cardiac pacemakers*. Regulations for protecting against malfunctions due to electromagnetic interference (corresponds with VDE 0750). http://www.etsi.org
EN 300 220: *Electromagnetic Compatibility and Radio Spectrum Matters (ERM); Short Range Devices (SRD); Radio equipment to be used in the 25MHz to 1000MHz frequency range with power levels ranging up to 500mW*. http://www.etsi.org
Part 1: Technical characteristics and test methods
Part 2: Supplementary parameters not intended for conformity purposes
Part 3: Harmonized EN covering essential requirements under article 3.2 of the R&TTE Directive
EN 300 330: *Electromagnetic Compatibility and Radio Spectrum Matters (ERM); Short Range Devices (SRD); Radio equipment in the frequency range 9 kHz to 25MHz and inductive loop systems in the frequency range 9 kHz to 30MHz*. http://www.etsi.org
Part 1: Technical characteristics and test methods
Part 2: Harmonized EN under article 3.2 of the R&TTE Directive.
EN 300 440: *Radio Equipment and Systems (RES); Short range devices, Technical characteristics and test methods for radio equipment to be used in the 1GHz to 25GHz frequency range with power levels ranging up to 500mW* . http://www.etsi.orgRadio Equipment and Systems (RES); Short range devicesRange Devices
ETS 300 683: *Radio Equipment and Systems (RES); ElectroMagnetic Compatibility (EMC) standard for Short Range Devices (SRD) operating on frequencies between 9 kHz and 25GHz*. http://www.etsi.org
EN 300 761: *Electromagnetic compatibility and Radio spectrum Matters (ERM); Short Range Devices (SRD); Automatic Vehicle Identification* (AVI) for *railways* operating in the 2.45 GHz frequency range. http://www.etsi.org
Part 1: Technical characteristics and methods of measurement.
Part 2: Harmonized standard covering essential requirements under article 3.2 of the R&TTE Directive
EN 300 674: *Electromagnetic ompatibility and Radio Spectrum Matters (ERM); Road Transport and Traffic Telematics (RTTT); Technical characteristics and test methods for dedicated short range communications (DSRC) transmission equipment (500*

kbit/s/250 kbit/s) operating in the 5.8GHz Industrial, Scientific and Medical (ISM) band. http://www.etsi.org
EN 301 489: *Electromagnetic Compatibility and Radio Spectrum Matters (ERM); Electromagnetic compatibility (EMC) standard for radio equipment and services*.
http://www.etsi.org
Part 1: Common technical requirements
Part 2: Specific requirements for radio paging equipment
Part 3: Specific requirements for short range devices (SRD) operating on frequencies between 9 kHz and 25 GHz
Part 4: Specific requirements for fixed radio links and ancillary equipment and services
Part 5: Specific requirements for private and mobile radio (PMR) and ancillary equipment (speech and non-speech)
Part 6: Specific conditions for digital enhanced cordless telecommunications (DECT) equipment
Part 7: Specific conditions for mobile and portable radio and ancillary equipment of digital cellular radio telecommunications systems (GSM and DCS)
Part 8: Specific requirements for GSM base stations
Part 9: Specific conditions for wireless microphones and similar radio frequency (RF) audio link equipment

Part 10: Specific conditions for first (CT1 and CT1 +) and second generation cordless telephone (CT2) equipment
Part 11: Specific conditions for FM broadcasting transmitters
Part 12: Specific conditions for Earth stations operated in the frequency ranges between 4 GHz and 30 GHz in the fixed satellite service (FSS)
Part 13: Specific conditions for citizens' band (CB) radio and ancillary equipment (speech and non-speech)
Part 15: Specific conditions for commercially available amateur radio equipment
Part 16: Specific conditions for analogue cellular radio communications equipment, mobile and portable
Part 17: Specific requirements for wideband data and HIPERLAN
Part 18: Specific requirements for terrestrial trunked radio (TETRA)
Part 19: Specific conditions for receive only mobile Earth stations (ROMES) operating in the 1.5 GHz band providing data communications
Part 20: Specific conditions for mobile Earth stations (MES) used in the mobile satellite services (MSS)
Part 22: Specific requirements for VHF aeronautical mobile and fixed radios
ERC/DEC 92-02: *CEPT/ERC Decision on the frequency bands to be designated for the coordinated introduction of road transport telematic systems*. http://www.ero.dk
ERC/DEC 97-10: *CEPT/ERC Decision on the mutual recognition of conformity assessment procedures including marking of radio equipment and radio terminal equipment*.
http://www.ero.dk
ERC/DEC 01-01: *CEPT/ERC Decision: non-specific short range devices in 6765–6795 kHz and 13.552–13.567MHz.* http://www.ero.dk
ERC/DEC 01-02: *CEPT/ERC Decision: non-specific short range devices in 26.957–27.283MHz.* http://www.ero.dk
ERC/DEC 01-03: *CEPT/ERC Decision: non-specific short range devices in 40.660–40.700MHz.* http://www.ero.dk
ERC/DEC 01-04: *CEPT/ERC Decision: non-specific short range devices in 868.0–868.6 MHz, 868.7–869.2 MHz, 869.4–869.65 MHz, 869.7–870.0MHz*. http://www.ero.dk
ERC/DEC 01-05: *CEPT/ERC Decision: non-specific short range devices in 2400–2483.5MHz.* http://www.ero.dk
ERC/DEC 01-13: *CEPT/ERC Decision: short range devices for inductive applications in 9–59,750 kHz, 59.750–60.250 kHz, 60.250–70 kHz, 70–119 kHz and 119–135 kHz.* http://www.ero.dk
ERC/DEC 01-14: *CEPT/ERC Decision: short range devices for inductive applications in 6765–6795 kHz, 13.553–13.567MHz*. http://www.ero.dk
ERC/DEC 01-15: *CEPT/ERC Decision: short range devices for inductive applications in 7400–8800 kHz*. http://www.ero.dk
ERC/DEC 01-16: *CEPT/ERC Decision: short range devices for inductive applications in 26.957–27.283MHz.* http://www.ero.dk
ERC/REC 01-06: *CEPT/ERC Recommendation: procedure for mutual recognition of type testing and type-approval for radio equipment*. http://www.ero.dk
ERC/REC 70-03: *CEPT/ERC Recommendation 70-03 relating to the use of short range devices (SRD).* http://www.ero.dk
ETSI TS 102 190: see ISO/IEC 18092 (NFCIP-1)
ETSI TS 102 312: see ISO/IEC 21481 (NFCIP-2)
ETSI TS 102 345: see ISO/IEC 22536 (NFCIP-1; RF interface test methods)
ISO/IEC 6346 Freight containers – coding, identification and marking
ISO/IEC 7810: Identification cards – physical characteristics
ISO/IEC 7816: Identification cards – integrated circuit(s) cards with contacts
Part 1: Physical characteristics
Part 2: Dimensions and location of the contacts

Part 3: Electronic signals and transmission protocols
Part 4: Interindustry commands for interchange

Part 5: Registration system for applications in IC cards
Part 6: Interindustry data elements
Part 7: Interindustry commands for structured card query language (SCQL)
Part 8: Security architecture and related interindustry commands
Part 9: Enhanced interindustry commands
Part 10: Electronic signals and answer to reset for synchronous cards
Part 11: Card structure and enhanced functions for multi-application use
Part 12: Cryptographic information application
ISO/IEC 8824-1: Information technology – Abstract Syntax Notation One (ASN.1) – specification of basic notation.
ISO/IEC 8825-1: Information technology – ASN.1 encoding rules – specification of basic encoding rules (BER), canonical encoding rules (CER) and distinguished encoding rules (DER).
ISO/IEC 9798: *Information technology – security techniques – entity authentication*. Principles and description of authentication procedures
Part 1: General
Part 2: Mechanisms using symmetric encipherment algorithms
Part 3: Mechanisms using digital signature techniques
Part 4: Mechanisms using a cryptographic check functions
Part 5: Mechanisms using zero knowledge techniques
ISO/IEC 9834-1: 1993/Amd.2: 1988 Information technology – open systems interconnection – procedures for the operation of OSI registration authorities: general procedures
ISO/IEC 10373: *Identification cards – test methods*. Test methods for 'plastic cards' for testing the card body and the fitted card element (magnetic strip, semiconductor chip). The standard consists of the following parts:
Part 1: General
Part 2: Magnetic strip technologies
Part 3: Integrated circuit cards (smart cards with contact)
Part 4: Contactless integrated circuit cards (close coupling)
Part 5: Optical memory cards
Part 6: Proximity cards (contactless smart cards acc. to ISO/IEC 14443)
Part 7: Vicinity cards (contactless smart cards acc. to ISO/IEC 15693)
ISO/IEC 10374: *Container – Automatische Identifizierung* (f*reight containers – automatic identification*). Automatic identification of freight containers by a 2.45 GHz transponder system.
ISO/IEC 10536: *Identification cards – contactless integrated circuit(s) cards*. Contactless smart cards in close coupling technology. The standard consists of the following parts:
Part 1: Physical characteristics
Part 2: Dimensions and location of coupling areas
Part 3: Electronic signals and reset procedures
Part 4: Answer to reset and transmission protocols
ISO/IEC 11784: *Radio frequency identification of animals – code structure*. Identification of animals by RFID systems. Description of the data structure.
ISO/IEC 11785: *Radio frequency identification of animals – technical concept*. Identification of animals by RFID systems. Description of the RF transmission procedure.
ISO/IEC 14223: *Radio frequency identification of animals – advanced transponders*:
Part 1: Air interface
Part 2: Code and command structure
ISO/IEC 14443: *Identification cards – proximity integrated circuit(s) cards*:
Part 1: Physical characteristics
Part 2: Radio frequency interface
Part 3: Initialization and anticollision
Part 4: Transmission protocols

ISO/IEC 14816: Road traffic and transport telematics – automatic vehicle and equipment identification – numbering and data structures
ISO/IEC 15459: Information technology – automatic identification and data capture techniques – unique identifiers for item management
Part 1: Unique identification of transport units
Part 2: Registration procedures
Part 3: Common rules for unique identification
Part 4: Unique item identification for supply chain management
Part 5: Unique identification of returnable transport items (RTIs)
Part 6: Unique identification for product groupings in material lifecyle management
ISO/IEC 15693: Identification cards – contactless integrated circuit(s) cards – vicinity cards
Part 1: Physical characteristics
Part 2: Air interface and initialisation
Part 3: Protocols
Part 4: Registration of applications/issuers
ISO/IEC 15961: Information technology – RFID for i*tem management* – Data protocol: application interface

ISO/IEC 15962: Information technology – RFID for *item management* – Data protocol: data encoding rules and logical memory functions
ISO/IEC 15963: Unique identification of RF tag and registration authority to manage the uniqueness
Part 1: Numbering system
Part 2: Procedural standard
Part 3: Use of the unique identification of RF tag in the integrated circuit
ISO/IEC 17358: Supply chain application for RFID – application requirements
ISO/IEC 17363: Supply chain application for RFID – freight containers
ISO/IEC 17364: Supply chain application for RFID – transport units
ISO/IEC 17365: Supply chain application for RFID – returnable transport items
ISO/IEC 17366: Supply chain application for RFID – product packaging
ISO/IEC 17367: Supply chain application for RFID – product tagging
ISO/IEC 18000: RFID for i*tem management* – air interface
Part 1: Generic parameter for air interface communication for globally accepted frequencies
Part 2: Parameters for air interface communication below 135 kHz
Part 3: Parameters for air interface communication at 13.56 MHz
Part 4: Parameters for air interface communication at 2.45 GHz
Part 5: Parameters for air interface communication at 5.8 GHz
Part 6: Parameters for air interface communication – UHF frequency band (868 / 915 MHz)
ISO/IEC 18001: Information technology – radio frequency identification for item management – application requirements profiles.
ISO/IEC 18046: RFID tag and interrogator performance test methods
ISO/IEC 18047: Information technology – radio frequency identification device conformance test methods – test methods for ISO/IEC 18000
Part 3: Test methods for air interface communications at 13.56 MHz
Part 4: Test methods for air interface communications at 2.45 GHz
Part 7: Test methods for air interface communications at 433 MHz
ISO/IEC 18092: Near field communication (NFC) interface and protocol-1 (NFCIP-1)
ISO/IEC 18185: Freight containers – radio frequency communication protocol for electronic seal
Part 1: Communication protocol
Part 2: Application requirements
Part 3: Environmental characteristics
Part 4: Data protection
Part 5: Sensor interface
Part 6: Message sets for transfer btw. seal reader and host computer
ISO/IEC 19762: Information technology AIDC techniques – harmonized vocabulary
Part 1: General terms relating to automatic identification and data capture (AIDC).
Part 2: Optically readable media (ORM)
Part 3: Radio frequency identification.
ISO/IEC 21007: Gas cylinders – identification and marking using radio frequency identification technology
Part 1: Reference architecture and terminology
Part 2: Numbering schemes for radio frequency
ISO/IEC 21481: Near Field Communication (NFC) Interface and Protocol-2 (NFCIP-2).
ISO/IEC 22536: Near Field Communication (NFC) Interface and Protocol-1 (NFCIP-1); RF Interface Test Methods
ISO/IEC 23389: Freight containers – read write radio frequency identification (RFID).
ISO/IEC 23917: Near Field Communication (NFC) Interface and Protocol-2 (NFCIP-2); Protocol Test Methods for NFC
ISO/IEC 24710: Information technology AIDC techniques – RFID for item management – ISO/IEC 18000 Air Interface Communications – elementary tag license-plate functionality for ISO/IEC 18000 air interface definitions
ISO/IEC 24729: Information technology – radio frequency identification for item management – implementation guidelines
Part 1: RFID-enabled labels and packaging
Part 2: Recyclability of RF tags
Part 3: RFID interrogator/antenna installation
ISO 69873: *Tools and clamping devices with data carriers – dimensions for data carriers and their fitting space*
S-918-00: AAR Manual of Standards and Recommended Practices Railway Electronics,
S-918: *Standard for Automatic Equipment Identification*. Adopted: 1991; Revised: 1995, 2000
VDE 0848: *Safety in electromagnetic fields (Part 2 – Protection of people in the frequency range 30 kHz to 300 GHz, Part 4A2 – Protection of people in the frequency range 0Hz – 30kHz*.
VDE 0750: See EN 50061.
VDI 4470 – Teil 1: *Warensicherungssysteme – Kundenabnahmerichtlinie f˙ur Schleusensysteme*. Onsite determination of the detection rate when EAS systems are put into operation.
VDI 4470 – Teil 2: *Warensicherungssysteme – Kundenabnahmerichtlinie f˙ur Deaktivierungsanlagen*. Testing of deactivation equipment for EAS systems.
Part 7: Physical layer

LFSR

An integrated Pseudo-Random Number Generator (PRNG), clocked simultaneously with the cipher, is utilized to generate nonces for the authentication between reader and card. It is based on a 16-bit LFSR with feedback polynomial $x16 + x14 + x13 + x11 + 1$.
The state of
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
SHIFT DIRECTION
the LFSR is expanded to 4-byte values by shifting the output bits of the 16-bit LFSR into an
106
6.3. Mifare Classic Cards
enlarged 32-bit shift-register, as illustrated in Fig. 6.3. Therefore, the resulting 32-bit pseudorandom numbers contain only 16 bit of entropy, i.e., only 216 = 65636 different numbers are generated. Our experiments confirm that the PRNG is clocked every 27 = 128 clock cycles, synchronously to the carrier frequency of 13.56MHz, hence the same number is output every $65536 \cdot 9.44\ \mu s \approx 619ms$.

Anexo código básico // Modificaciones para añadir board IMXT

```
//////////////////////
#ifdef PHDRIVER_KINETIS_K82
#   include <fsl_debug_console.h>
#   include <stdio.h>
#endif

#ifdef PHDRIVER_IMXRT_1050
#   include <fsl_debug_console.h>
#   include <stdio.h>
#endif
//////////////////////////////////////////////////////
/* Check for K82 controller based boards. */
#if             defined(PHDRIVER_FRDM_K82FPN5180_BOARD)             ||
defined(PHDRIVER_FRDM_K82FRC663_BOARD)
#define PHDRIVER_KINETIS_K82
#endif

/* Check for LPC1769 controller based boards. */
#if              defined(PHDRIVER_LPC1769PN5180_BOARD)              ||
defined(PHDRIVER_LPC1769RC663_BOARD)
#define PHDRIVER_LPC1769
#endif
/* Añadimos IMXRT1050.
 */
 #if defined(PHDRIVER_IMXRT1050RC663_BOARD)
#define PHDRIVER_IMXRT_1050
#endif
//////////////////////////////////////////////////////
 #if defined PHDRIVER_KINETIS_K82
    phApp_K82_Init();
#elif defined NXPBUILD__PHHAL_HW_PN7462AU
    phFlashBoot_Main();

    phhalTimer_Init();
#elif defined(PHDRIVER_LPC1769) && defined(__CC_ARM)
    SystemCoreClock =  (( unsigned long ) 96000000);
#elif defined(PHDRIVER_IMXRT_1050)  // Definimos
      phApp_IMXRT1050_Init();

      ////////////////////////////////////////////////////////////////
#if defined PHDRIVER_KINETIS_K82
    phApp_K82_Init();
#elif defined NXPBUILD__PHHAL_HW_PN7462AU
    phFlashBoot_Main();

    phhalTimer_Init();
#elif defined(PHDRIVER_LPC1769) && defined(__CC_ARM)
    SystemCoreClock =  (( unsigned long ) 96000000);
#elif defined(PHDRIVER_IMXRT_1050)  // Definimos
      phApp_IMXRT1050_Init();

////////////////////////////////////////////////////////////////

      #ifdef PHDRIVER_KINETIS_K82
    NVIC_SetPriority(EINT_IRQn, EINT_PRIORITY);
    NVIC_ClearPendingIRQ(EINT_IRQn);
    EnableIRQ(EINT_IRQn);
#endif /* PHDRIVER_KINETIS_K82 */

#ifdef PHDRIVER_IMXRT_1050
```

```
    NVIC_SetPriority(EINT_IRQn, EINT_PRIORITY);
    NVIC_ClearPendingIRQ(EINT_IRQn);
    EnableIRQ(EINT_IRQn);
#endif /* PHDRIVER_KINETIS_K82 */



  ////////////////////En el directorio NXP_Build
////////////////////////
  #if defined(PHDRIVER_LPC1769RC663_BOARD) \
    || defined(PHDRIVER_FRDM_K82FRC663_BOARD)\
      || defined(PHDRIVER_IMXRT1050RC663_BOARD)    /* Añadimos nuestra
board */

#   define NXPBUILD__PHHAL_HW_RC663
#endif
```

Anexo código básico // Codigo Main NFC Discovery Loop

```
//////////////////////////CODIGO COMPLETO ///////////////////////
#include <phApp_Init.h>

/* Local headers */
#include "NfcrdlibEx1_BasicDiscoveryLoop.h"

/*****************************************************************
**********
**   Global Defines
*****************************************************************
*********/

phacDiscLoop_Sw_DataParams_t       * pDiscLoop;       /* Discovery loop
component */

/*The below variables needs to be initialized according to example
requirements by a customer */

uint8_t  sens_res[2]      = {0x04, 0x00};               /* ATQ bytes -
needed for anti-collision */
uint8_t  nfc_id1[3]       = {0xA1, 0xA2, 0xA3};         /* user defined
bytes of the UID (one is hardcoded) - needed for anti-collision */
uint8_t  sel_res         = 0x40;
uint8_t  nfc_id3          = 0xFA;                       /* NFC3 byte -
required for anti-collision */
uint8_t  poll_res[18]     = {0x01, 0xFE, 0xB2, 0xB3, 0xB4, 0xB5,
                             0xB6, 0xB7, 0xC0, 0xC1, 0xC2, 0xC3,
                             0xC4, 0xC5, 0xC6, 0xC7, 0x23, 0x45
};

#ifdef PHOSAL_FREERTOS_STATIC_MEM_ALLOCATION
uint32_t aBasicDiscTaskBuffer[BASIC_DISC_DEMO_TASK_STACK];
#else /* uint32_t aBasicDiscTaskBuffer[BASIC_DISC_DEMO_TASK_STACK]; */
#define aBasicDiscTaskBuffer    NULL
#endif /* PHOSAL_FREERTOS_STATIC_MEM_ALLOCATION */
```

```
/*****************************************************************
**********
**   Static Defines
*****************************************************************
*********/

/* This is used to save restore Poll Config.
 * If in case application has update/change PollCfg to resolve Tech
 * when Multiple Tech was detected in previous poll cycle
 */
static uint16_t bSavePollTechCfg;

/*****************************************************************
**********
**   Function Declarations
*****************************************************************
*********/
void BasicDiscoveryLoop_Demo(void  *pDataParams);

/*****************************************************************
**********
**   Function Definitions
*****************************************************************
*********/


/*****************************************************************
**********
**   Main Function
*****************************************************************
*********/

int main (void)
{
    do
    {
        phStatus_t             status = PH_ERR_INTERNAL_ERROR;
        phNfcLib_Status_t      dwStatus;

#ifdef PH_PLATFORM_HAS_ICFRONTEND
        phNfcLib_AppContext_t AppContext = {0};
#endif /* PH_PLATFORM_HAS_ICFRONTEND */

#ifndef PH_OSAL_NULLOS
        phOsal_ThreadObj_t BasicDisc;
#endif /* PH_OSAL_NULLOS */

        /* Perform Controller specific initialization. */
        phApp_CPU_Init();

        /* Perform OSAL Initialization. */
        (void)phOsal_Init();

        DEBUG_PRINTF("\n\r BasicDiscoveryLoop Example: \n\r");

#ifdef PH_PLATFORM_HAS_ICFRONTEND
        status = phbalReg_Init(&sBalParams, sizeof(phbalReg_Type_t));
        CHECK_STATUS(status);
```

```
        AppContext.pBalDataparams = &sBalParams;
        dwStatus = phNfcLib_SetContext(&AppContext);
        CHECK_NFCLIB_STATUS(dwStatus);
#endif

        /* Initialize library */
        dwStatus = phNfcLib_Init();
        CHECK_NFCLIB_STATUS(dwStatus);
        if(dwStatus != PH_NFCLIB_STATUS_SUCCESS) break;

        /* Set the generic pointer */
        pHal = phNfcLib_GetDataParams(PH_COMP_HAL);
        pDiscLoop = phNfcLib_GetDataParams(PH_COMP_AC_DISCLOOP);

        /* Initialize other components that are not initialized by NFCLIB
and configure Discovery Loop. */
        status = phApp_Comp_Init(pDiscLoop);
        CHECK_STATUS(status);
        if(status != PH_ERR_SUCCESS) break;

        /* Configure the IRQ */
        status = phApp_Configure_IRQ();
        CHECK_STATUS(status);
        if(status != PH_ERR_SUCCESS) break;

#ifndef PH_OSAL_NULLOS

        BasicDisc.pTaskName = (uint8_t *) "BasicDiscLoop";
        BasicDisc.pStackBuffer = aBasicDiscTaskBuffer;
        BasicDisc.priority = BASIC_DISC_DEMO_TASK_PRIO;
        BasicDisc.stackSizeInNum = BASIC_DISC_DEMO_TASK_STACK;
        phOsal_ThreadCreate(&BasicDisc.ThreadHandle,        &BasicDisc,
&BasicDiscoveryLoop_Demo, pDiscLoop);
        phOsal_StartScheduler();
        DEBUG_PRINTF("RTOS Error : Scheduler exited. \n\r");

#else
        (void)BasicDiscoveryLoop_Demo(pDiscLoop);
#endif
    } while(0);

    while(1); //Comes here if initialization failure or scheduler exit
due to error

    return 0;
}

/**
* This function demonstrates the usage of discovery loop
* It detects and reports the NFC technology type
* \param   pDataParams      The discovery loop data parameters
* \note    This function will never return
*/
void BasicDiscoveryLoop_Demo(void  *pDataParams)
{
    phStatus_t    status, statustmp;
    uint16_t      wTagsDetected = 0;
    uint16_t      wNumberOfTags = 0;
    uint16_t      wEntryPoint;
    uint16_t      wValue;
```

```
    uint8_t       bIndex;

    status = phApp_HALConfigAutoColl();
    CHECK_STATUS(status);

    /* Get Poll Configuration */
    status              =             phacDiscLoop_GetConfig(pDataParams,
PHAC_DISCLOOP_CONFIG_PAS_POLL_TECH_CFG, &bSavePollTechCfg);
    CHECK_STATUS(status);

    /* Start in poll mode */
    wEntryPoint = PHAC_DISCLOOP_ENTRY_POINT_POLL;
    status = PHAC_DISCLOOP_LPCD_NO_TECH_DETECTED;

    while(1)
    {
        /* Switch off RF field */
        statustmp = phhalHw_FieldOff(pHal);
        CHECK_STATUS(statustmp);

        /* Set Discovery Poll State to Detection */
        statustmp         =         phacDiscLoop_SetConfig(pDataParams,
PHAC_DISCLOOP_CONFIG_NEXT_POLL_STATE,
PHAC_DISCLOOP_POLL_STATE_DETECTION);
        CHECK_STATUS(statustmp);

        /* Set Poll Configuration */
        statustmp         =         phacDiscLoop_SetConfig(pDataParams,
PHAC_DISCLOOP_CONFIG_PAS_POLL_TECH_CFG, bSavePollTechCfg);
        CHECK_STATUS(statustmp);

#ifdef PH_EXAMPLE1_LPCD_ENABLE

#ifdef NXPBUILD__PHHAL_HW_RC663
        if (wEntryPoint == PHAC_DISCLOOP_ENTRY_POINT_POLL)
#else
        /* Configure LPCD */
        if          ((status          &          PH_ERR_MASK)          ==
PHAC_DISCLOOP_LPCD_NO_TECH_DETECTED)
#endif
        {
            status = phApp_ConfigureLPCD();
            CHECK_STATUS(status);
        }

        /* Bool to enable LPCD feature. */
        status          =          phacDiscLoop_SetConfig(pDataParams,
PHAC_DISCLOOP_CONFIG_ENABLE_LPCD, PH_ON);
        CHECK_STATUS(status);
#endif /* PH_EXAMPLE1_LPCD_ENABLE */

        /* Start discovery loop */
        status = phacDiscLoop_Run(pDataParams, wEntryPoint);

        if(wEntryPoint == PHAC_DISCLOOP_ENTRY_POINT_POLL)
        {
            if((status            &            PH_ERR_MASK)            ==
PHAC_DISCLOOP_MULTI_TECH_DETECTED)
            {
```

```
                DEBUG_PRINTF  ("  \n\r  Multiple  technology  detected:
\n\r");

                status      =      phacDiscLoop_GetConfig(pDataParams,
PHAC_DISCLOOP_CONFIG_TECH_DETECTED, &wTagsDetected);
                CHECK_STATUS(status);

                if(PHAC_DISCLOOP_CHECK_ANDMASK(wTagsDetected,
PHAC_DISCLOOP_POS_BIT_MASK_A))
                {
                    DEBUG_PRINTF (" \tType A detected... \n\r");
                }
                if(PHAC_DISCLOOP_CHECK_ANDMASK(wTagsDetected,
PHAC_DISCLOOP_POS_BIT_MASK_B))
                {
                    DEBUG_PRINTF (" \tType B detected... \n\r");
                }
                if(PHAC_DISCLOOP_CHECK_ANDMASK(wTagsDetected,
PHAC_DISCLOOP_POS_BIT_MASK_F212))
                {
                    DEBUG_PRINTF  (" \tType  F  detected  with  baud  rate
212... \n\r");
                }
                if(PHAC_DISCLOOP_CHECK_ANDMASK(wTagsDetected,
PHAC_DISCLOOP_POS_BIT_MASK_F424))
                {
                    DEBUG_PRINTF  (" \tType  F  detected  with  baud  rate
424... \n\r");
                }
                if(PHAC_DISCLOOP_CHECK_ANDMASK(wTagsDetected,
PHAC_DISCLOOP_POS_BIT_MASK_V))
                {
                    DEBUG_PRINTF("  \tType  V  /  ISO  15693  /  T5T
detected... \n\r");
                }

                /* Select 1st Detected Technology to Resolve*/
                for(bIndex        =        0;        bIndex        <
PHAC_DISCLOOP_PASS_POLL_MAX_TECHS_SUPPORTED; bIndex++)
                {
                    if(PHAC_DISCLOOP_CHECK_ANDMASK(wTagsDetected, (1 <<
bIndex)))
                    {
                        /* Configure for one of the detected technology
*/
                        status  =  phacDiscLoop_SetConfig(pDataParams,
PHAC_DISCLOOP_CONFIG_PAS_POLL_TECH_CFG, (1 << bIndex));
                        CHECK_STATUS(status);
                        break;
                    }
                }

                /* Print the technology resolved */
                phApp_PrintTech((1 << bIndex));

                /* Set Discovery Poll State to collision resolution */
                status      =      phacDiscLoop_SetConfig(pDataParams,
PHAC_DISCLOOP_CONFIG_NEXT_POLL_STATE,
PHAC_DISCLOOP_POLL_STATE_COLLISION_RESOLUTION);
                CHECK_STATUS(status);
```

```
                /* Restart discovery loop in poll mode from collision
resolution phase */
                status = phacDiscLoop_Run(pDataParams, wEntryPoint);
            }

            if((status & PH_ERR_MASK) ==
PHAC_DISCLOOP_MULTI_DEVICES_RESOLVED)
            {
                /* Get Detected Technology Type */
                status = phacDiscLoop_GetConfig(pDiscLoop,
PHAC_DISCLOOP_CONFIG_TECH_DETECTED, &wTagsDetected);
                CHECK_STATUS(status);

                /* Get number of tags detected */
                status = phacDiscLoop_GetConfig(pDiscLoop,
PHAC_DISCLOOP_CONFIG_NR_TAGS_FOUND, &wNumberOfTags);
                CHECK_STATUS(status);

                DEBUG_PRINTF (" \n\r Multiple cards resolved: %d cards
\n\r",wNumberOfTags);
                phApp_PrintTagInfo(pDataParams, wNumberOfTags,
wTagsDetected);

                if(wNumberOfTags > 1)
                {
                    /* Get 1st Detected Tag and Activate device at index
0 */
                    for(bIndex = 0; bIndex <
PHAC_DISCLOOP_PASS_POLL_MAX_TECHS_SUPPORTED; bIndex++)
                    {
                        if(PHAC_DISCLOOP_CHECK_ANDMASK(wTagsDetected,
(1 << bIndex)))
                        {
                            DEBUG_PRINTF("\t Activating one
card...\n\r");
                            status =
phacDiscLoop_ActivateCard(pDataParams, bIndex, 0);
                            break;
                        }
                    }

                    if(((status & PH_ERR_MASK) ==
PHAC_DISCLOOP_DEVICE_ACTIVATED) ||
                            ((status & PH_ERR_MASK) ==
PHAC_DISCLOOP_PASSIVE_TARGET_ACTIVATED) ||
                            ((status & PH_ERR_MASK) ==
PHAC_DISCLOOP_MERGED_SEL_RES_FOUND))
                    {
                        /* Get Detected Technology Type */
                        status = phacDiscLoop_GetConfig(pDiscLoop,
PHAC_DISCLOOP_CONFIG_TECH_DETECTED, &wTagsDetected);
                        CHECK_STATUS(status);

                        phApp_PrintTagInfo(pDataParams, 0x01,
wTagsDetected);
                    }
                    else
                    {
```

```
                        PRINT_INFO("\t\tCard                    activation
failed...\n\r");
                    }
                }
                /* Switch to LISTEN mode after POLL mode */
            }
            else      if      (((status      &      PH_ERR_MASK)      ==
PHAC_DISCLOOP_NO_TECH_DETECTED) ||
                    ((status          &          PH_ERR_MASK)          ==
PHAC_DISCLOOP_NO_DEVICE_RESOLVED))
            {
                /* Switch to LISTEN mode after POLL mode */
            }
            else        if((status        &        PH_ERR_MASK)        ==
PHAC_DISCLOOP_EXTERNAL_RFON)
            {
                /*
                 * If external RF is detected during POLL, return back
so that the application
                 * can restart the loop in LISTEN mode
                 */
            }
            else        if((status        &        PH_ERR_MASK)        ==
PHAC_DISCLOOP_MERGED_SEL_RES_FOUND)
            {
                DEBUG_PRINTF  ("  \n\r  Device  having  T4T  and  NFC-DEP
support detected... \n\r");

                /* Get Detected Technology Type */
                status        =        phacDiscLoop_GetConfig(pDiscLoop,
PHAC_DISCLOOP_CONFIG_TECH_DETECTED, &wTagsDetected);
                CHECK_STATUS(status);

                phApp_PrintTagInfo(pDataParams, 1, wTagsDetected);

                /* Switch to LISTEN mode after POLL mode */
            }
            else        if((status        &        PH_ERR_MASK)        ==
PHAC_DISCLOOP_DEVICE_ACTIVATED)
            {
                DEBUG_PRINTF  ("  \n\r  Card  detected  and  activated
successfully... \n\r");
                status       =       phacDiscLoop_GetConfig(pDataParams,
PHAC_DISCLOOP_CONFIG_NR_TAGS_FOUND, &wNumberOfTags);
                CHECK_STATUS(status);

                /* Get Detected Technology Type */
                status       =       phacDiscLoop_GetConfig(pDataParams,
PHAC_DISCLOOP_CONFIG_TECH_DETECTED, &wTagsDetected);
                CHECK_STATUS(status);

                phApp_PrintTagInfo(pDataParams,          wNumberOfTags,
wTagsDetected);

                /* Switch to LISTEN mode after POLL mode */
            }
            else        if((status        &        PH_ERR_MASK)        ==
PHAC_DISCLOOP_ACTIVE_TARGET_ACTIVATED)
            {
                DEBUG_PRINTF (" \n\r Active target detected... \n\r");
```

```
                /* Switch to LISTEN mode after POLL mode */
            }
            else        if((status        &        PH_ERR_MASK)        ==
PHAC_DISCLOOP_PASSIVE_TARGET_ACTIVATED)
            {
                DEBUG_PRINTF (" \n\r Passive target detected... \n\r");

                /* Get Detected Technology Type */
                status        =        phacDiscLoop_GetConfig(pDiscLoop,
PHAC_DISCLOOP_CONFIG_TECH_DETECTED, &wTagsDetected);
                CHECK_STATUS(status);

                phApp_PrintTagInfo(pDataParams, 1, wTagsDetected);

                /* Switch to LISTEN mode after POLL mode */
            }
            else      if      ((status      &      PH_ERR_MASK)      ==
PHAC_DISCLOOP_LPCD_NO_TECH_DETECTED)
            {
                /* LPCD is succeed but no tag is detected. */
            }
            else
            {
                if((status & PH_ERR_MASK) == PHAC_DISCLOOP_FAILURE)
                {
                    status     =     phacDiscLoop_GetConfig(pDataParams,
PHAC_DISCLOOP_CONFIG_ADDITIONAL_INFO, &wValue);
                    CHECK_STATUS(status);
                    DEBUG_ERROR_PRINT(PrintErrorInfo(wValue));
                }
                else
                {
                    DEBUG_ERROR_PRINT(PrintErrorInfo(status));
                }
            }

            /* Update the Entry point to LISTEN mode. */
            wEntryPoint = PHAC_DISCLOOP_ENTRY_POINT_LISTEN;
        }
        else
        {
            if((status & PH_ERR_MASK) == PHAC_DISCLOOP_EXTERNAL_RFOFF)
            {
                /*
                 * Enters here if in the target/card mode and external
RF is not available
                 *  Wait  for  LISTEN  timeout  till  an  external  RF  is
detected.
                 * Application may choose to go into standby at this
point.
                 */
                status = phhalHw_EventConsume(pHal);
                CHECK_STATUS(status);

                status            =            phhalHw_SetConfig(pHal,
PHHAL_HW_CONFIG_RFON_INTERRUPT, PH_ON);
                CHECK_STATUS(status);

                status = phhalHw_EventWait(pHal, LISTEN_PHASE_TIME_MS);
                if((status & PH_ERR_MASK) == PH_ERR_IO_TIMEOUT)
```

```
                {
                    wEntryPoint = PHAC_DISCLOOP_ENTRY_POINT_POLL;
                }
                else
                {
                    wEntryPoint = PHAC_DISCLOOP_ENTRY_POINT_LISTEN;
                }
            }
            else
            {
                if((status & PH_ERR_MASK) == 
PHAC_DISCLOOP_ACTIVATED_BY_PEER)
                {
                    DEBUG_PRINTF (" \n\r Device activated in listen 
mode... \n\r");
                }
                else if ((status & PH_ERR_MASK) == 
PH_ERR_INVALID_PARAMETER)
                {
                    /* In case of Front end used is RC663, then listen 
mode is not supported.
                     * Switch from listen mode to poll mode. */
                }
                else
                {
                    if((status & PH_ERR_MASK) == PHAC_DISCLOOP_FAILURE)
                    {
                        status = phacDiscLoop_GetConfig(pDataParams, 
PHAC_DISCLOOP_CONFIG_ADDITIONAL_INFO, &wValue);
                        CHECK_STATUS(status);
                        DEBUG_ERROR_PRINT(PrintErrorInfo(wValue));
                    }
                    else
                    {
                        DEBUG_ERROR_PRINT(PrintErrorInfo(status));
                    }
                }

                /* On successful activated by Peer, switch to LISTEN 
mode */
                wEntryPoint = PHAC_DISCLOOP_ENTRY_POINT_POLL;
            }
        }
    }
}
```

6 Bibliografía

Bibliografía

102190, E. T. (s.f.). *https://www.etsi.org/deliver/etsi_ts/102100_102199/102190/01.01.01_60/ts_102190v010101p.pdf.*

Arnaud, M. (2011). *RFID Security and Privacy.* Paris, Francia: University Paris Ouest Nanterre.

Breitfuß, E. H. (s.f.). *Security in Near Field Communication - Strengths and Weaknesses.* Gratkorn, Austria: Philips Semiconductors.

Dennis Giese, K. L. (2018). *Security Analysis of Near-Field Communication (NFC).*

Finkenzeller, K. (2010). *RFID Handbook* (Tercera ed.). United Kingdom: Wiley.

GlobalPlatform. (2009). *Secure Channel Protocol 03.* Obtenido de https://globalplatform.org/wp-content/uploads/2019/03/GPC_2.2_D_SCP03_v1.0.pdf

Houda Ferradi, R. G. (2010). *A Forensic Analysis of an In-Card Listening Device.* Gardanne, France: Computer Science Department.

Kasper, T. (2012). *SECURITY ANALYSIS OF PERVASIVE WIRELESS DEVICE, PhD Thesis.* Bochum, Germany.

Milanov, E. (2009). *The RSA Algorithm.*

Motlagh, N. H. (2012). *NFC - A technical Overview.* THES.

N.a. (2011). *Near Field Communication.* Rohde & Schwarz.

Peter Darcy, P. P. (s.f.). *The Challenges and Issues Facing the Deploument of RFID.* Australia: Institute of Integrated and Intelligent Systems, Griffith University.

Sanjay Ahuja, P. P. (2010). *RFID Technolgy.* Jacksonville, Florida: School of Computing, University of North Florida.

Schneider, B. (1996). *Applied Cryptography, Protocols, Algorthms and Source Code.*

Sonal Rohilla, Syscom Corporation. (2015). *Secure Element, An evolution to existing secure technology.* Morpho,Safran: International Journal of Scientific and Research Publications, Volumen 5, I-7.

Want, R. (2006). Introduction to RFID Technology. *Pervasive Computing*, 25-34.

https://www.twistedtraces.com/online-quote# // Calculadora de Precios de PCB

https://www.commoncriteriaportal.org/files/epfiles/0944b_pdf.pdf

https://www.theverge.com/2020/6/22/21299182/apple-carkey-ios-14-13-digital-key-unlock-car-iphone-wwdc-2020

Proyecto Metro de Madrid

http://www.madrid.org/contratos-publicos/1354768366075/1109266180653/1354768360079.pdf

http://www.madrid.org/contratos-publicos/1354728608136/1350930820359/1354728614536.pdf